

第一本无人驾驶 技术书

刘少山 唐 洁 著
吴 双 李力耘



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

第一本无人驾驶 技术书

刘少山 唐 洁 著
吴 双 李力耘

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING
5rj.s.cn □□□□□□

内 容 简 介

无人驾驶是一个复杂的系统,涉及的技术点种类多且跨度大,入门者常常不知从何入手。本书首先宏观地呈现了无人驾驶的整体技术架构,概述了无人驾驶中涉及的各项技术点。在读者对无人驾驶技术有了宏观认识后,本书深入浅出地讲解了无人驾驶定位导航、感知、决策与控制等算法,深度学习在无人驾驶中的应用,无人驾驶系统软件和硬件平台,无人驾驶安全及无人驾驶云平台等多个主要技术点。本书的作者都是无人驾驶行业的从业者与研究人员,有着多年无人驾驶及人工智能技术的实战经验。

本书从实用的角度出发,以期帮助对无人驾驶技术(特别是人工智能在无人驾驶中的应用)感兴趣的从业者与相关人士实现对无人驾驶行业的快速入门,以及对无人驾驶技术的深度理解与应用实践。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

第一本无人驾驶技术书 / 刘少山等著. —北京:电子工业出版社, 2017.6
ISBN 978-7-121-31355-4

I. ①第… II. ①刘… III. ①人工智能—算法—研究 IV. ①TP18

中国版本图书馆 CIP 数据核字(2017)第 077646 号

策划编辑:郑柳洁

责任编辑:郑柳洁

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:720×1000 1/16 印张:13.75 字数:261 千字

版 次:2017 年 6 月第 1 版

印 次:2017 年 6 月第 1 次印刷

定 价:59.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:010-51260888-819, faq@phei.com.cn。



好评袭来

本书深入浅出地呈现了无人驾驶这个复杂的系统。书中包括无人驾驶定位与感知算法、无人驾驶决策与控制算法、深度学习在无人驾驶中的应用、无人驾驶系统、无人驾驶云平台、无人驾驶安全等章节，既宏观地呈现了无人驾驶技术的架构，又很好地深入到无人驾驶涉及的每个技术点。我相当同意书中的观点：无人驾驶并不是一个技术点，而是众多技术点的集合。无人车上路行驶的前提是每一个技术点都要做得很好，这就代表在每个技术点上都有很好的创新机会。例如，在无人驾驶芯片的设计上，使用低能耗的 ARM 架构加上不同的加速芯片（GPU、FPGA、DSP、ASIC）在性能与能耗上有很大优势。我仔细读完本书后，对整个无人驾驶系统架构有了很好的认识，当我想更深入地了解一个技术点时，本书也提供了很好的文献信息让我深入学习。毫不夸张地说，本书让我在短时间内对无人驾驶技术有了很好的了解。

ARM 董事会成员、全球副总裁、中国区总裁 吴昂雄

作为一名科技行业的从业者，我有幸近距离观察了许多所谓的颠覆性技术的生命周期。我的感受是人们会将一项技术的近期作用无限夸大，对这项技术的长期演化往往估计不足。这一次大家对“无人驾驶”的态度也一样，大多数人仅仅把“无人驾驶”看成一项技术，认为只要搞定算法、搞定传感器、搞定云与端的传输等就万事大吉，我们就进入了完全自动驾驶的时代。我认为这种想法很危险，首先，这是一种发明家而非创新家、投机者而非创业者的心态；其次，这些人没有充分认识到“无人驾驶”有着极大的安全属性与社会属性，他们可能会谈到政府管制的问题，但很少考虑人口结构及其背后的消费心理及消费习

惯问题。这种拿着锤子找钉子的做法对“无人驾驶”的落地很不利。

作为从业者之一，我坚定看好“无人驾驶”这一产业的长期趋势及其巨大的经济与社会效益，但在短期内，我们除了做好技术准备外，更应该把眼光放长远，虚心地研究市场、研究用户、研究监管者、研究利益相关方，脚踏实地一步一个脚印，共同实现“无人驾驶”这一可预见的未来。本书是我读过的有关“无人驾驶”最系统、最严谨的著作，值得有志于从事此行业的朋友认真阅读。

舜宇光学科技（集团）有限公司总裁、执行董事 孙决

刘少山带领的是一个专业而高效的硅谷精英团队！感谢他们的努力，将神秘高端的无人驾驶技术拉下神坛，并以庖丁解牛般的专业功底逐层剖析。这本诞生于工业界的无人驾驶图书，将极大地缩短开发者、爱好者，以及相关人士迅速切入，并深入学习和投身于人工智能无人驾驶这一热点领域的进程，实属可贵。

CSDN &《程序员》总编 孟迎霞

很高兴看到本书书稿，我认为这是一本无人驾驶方面的专业书籍，对技术发展现状和工业实现都进行了很好的描述，并对未来做出了展望。书中内容包括了各个层面面临的技术挑战和可能的技术解决方案，特别是在决策控制部分有精彩的描述。我相信本书对在校学生、研究生，以及工业界相关技术人员都有所帮助！

清华大学教授、博士生导师 樊平毅

当前，人工智能引起了全球性的关注，是一个可能改变世界的创新技术。无人驾驶技术是人工智能领域最重要的分支之一，其涉及的学科众多，是一个融会了大量新技术的工程实践。本书试图揭开无人驾驶技术的神秘面纱，使读者能够很快建立对无人驾驶技术的全面认识。本书从工程师的角度出发，全面介绍了无人驾驶技术涉及的核心方向，包括环境感知、车载传感器、规划控制，等等。书中涉及无人驾驶的多个技术方向自成体系，针对每个方向中的核心内容讨论了系统的技术思路和解决方案，在很多重要的技术上给出了颇具深度的细节示例。本书作者有深厚的硅谷工程师背景，作者描述的无人驾驶技术已经不是象牙塔里的学术，而是贴近社会并即将走进大众生活的新一代科技产品的实践。

电子科技大学教授、博士生导师 雷维礼



目录

1	无人车：正在开始的未来	1
1.1	正在走来的无人驾驶.....	2
1.2	自动驾驶的分级.....	4
1.3	无人驾驶系统简介.....	7
1.4	序幕刚启.....	18
1.5	参考资料.....	18
2	光学雷达在无人驾驶技术中的应用	21
2.1	无人驾驶技术简介.....	21
2.2	光学雷达基础知识.....	22
2.3	LiDAR 在无人驾驶技术中的应用领域.....	24
2.4	LiDAR 技术面临的挑战.....	26
2.5	展望未来.....	28
2.6	参考资料.....	28
3	GPS 及惯性传感器在无人驾驶中的应用	30
3.1	无人驾驶定位技术.....	30

3.2	GPS 简介	31
3.3	惯性传感器简介	34
3.4	GPS 和惯性传感器的融合	36
3.5	结论	37
3.6	参考资料	38

4 基于计算机视觉的无人驾驶感知系统 39

4.1	无人驾驶的感知	39
4.2	KITTI 数据集	40
4.3	计算机视觉能帮助无人车解决的问题	42
4.4	Optical Flow 和立体视觉	43
4.5	物体的识别与追踪	45
4.6	视觉里程计算法	47
4.7	结论	48
4.8	参考资料	49

5 卷积神经网络在无人驾驶中的应用 50

5.1	CNN 简介	50
5.2	无人驾驶双目 3D 感知	51
5.3	无人驾驶物体检测	54
5.4	结论	59
5.5	参考资料	59

6 增强学习在无人驾驶中的应用 61

6.1	增强学习简介	61
6.2	增强学习算法	63
6.3	使用增强学习帮助决策	68
6.4	无人驾驶的决策介绍	70
6.5	参考资料	74

7 无人驾驶的规划与控制 75

7.1 规划与控制简介	75
7.2 路由寻径	77
7.3 行为决策	84
7.4 动作规划	93
7.5 反馈控制	101
7.6 无人车规划控制结语	105
7.7 参考资料	105

8 基于 ROS 的无人驾驶系统 108

8.1 无人驾驶：多种技术的集成	108
8.2 机器人操作系统（ROS）简介	110
8.3 系统可靠性	115
8.4 系统通信性能提升	116
8.5 系统资源管理与安全性	117
8.6 结论	118
8.7 参考资料	118

9 无人驾驶的硬件平台 120

9.1 无人驾驶：复杂系统	120
9.2 传感器平台	121
9.3 计算平台	140
9.4 控制平台	150
9.5 结论	157
9.6 参考资料	158

10 无人驾驶系统安全 160

10.1 针对无人驾驶的安全威胁	160
10.2 无人驾驶传感器的安全	160
10.3 无人驾驶操作系统的安全	162

10.4	无人驾驶控制系统的安全	163
10.5	车联网通信系统的安全性	165
10.6	安全模型校验方法	168
10.7	参考资料	169

11 基于 Spark 与 ROS 的分布式无人驾驶模拟平台 171

11.1	无人驾驶模拟技术	171
11.2	基于 ROS 的无人驾驶模拟器	173
11.3	基于 Spark 的分布式的模拟平台	175
11.4	结论	178
11.5	参考资料	178

12 无人驾驶中的高精度地图 180

12.1	电子地图分类	180
12.2	高精度地图的特点	183
12.3	高精度地图的生产	185
12.4	无人驾驶场景中的应用	188
12.5	高精度地图的现状与结论	190
12.6	参考资料	191

13 无人驾驶的未来 192

13.1	无人驾驶的商业前景	192
13.2	无人车面临的障碍	194
13.3	无人驾驶产业	198
13.4	全球化下的无人驾驶	203
13.5	无人驾驶发展对策	205
13.6	可预见的未来	207
13.7	参考资料	208

1

无人车：正在开始的未来

我们已经拉开了全自动无人驾驶的序幕，在幕布之后精彩的未来将如何，让我们先回顾一下硅谷的发展历史，再以此展望无人驾驶的未来。如图 1-1 所示，现代信息技术始于 20 世纪 60 年代，仙童电子和 Intel 通过硅晶体微处理器技术的创新开创了信息技术的新时代，这也是硅谷的起源。微处理器技术极大地提高了工业化生产力，推进了现代工业的发展。20 世纪 80 年代，随着 Xerox Alto、Apple Lisa 及 Microsoft Windows 等软件系统的发展，图形界面被广泛应用，个人电脑的概念出现并开始普及，现代信息技术以此为基础普惠众人。



图 1-1 现代信息技术发展史

21 世纪初，在个人电脑逐步普及并被大规模应用的背景下，Google 的出现通过互联网和搜索引擎的方式将人与浩瀚如星海的信息互联起来了，至此，现代信息技术发展到了第三阶段。始于 2004 年的 Facebook 通过革新的社交网络模式将现代信息技术推进到了第四阶段。至此，人类的交往互联方式从线下扩展到了线上，人类社会在万维网上有了初始的迁移并逐步地成熟完善。

随着互联网人口规模的膨胀，Airbnb 与 Uber 等公司通过共享经济的思维把人类社会的经济模式直接推广到了互联网社会，利用互联网+移动设备等直接连接不同用户的经济行为，得到了大范围的成功。信息技术每一阶段的发展及其随后驱动的革新，都极大地改变了人类对信息的访问需求和获取方式。尤其对后几个阶段而言，互联网是一个基础性条件，大多数的服务是通过互联网传达给终端用户的。

现在，我们走到了信息技术发展的第六阶段，机器人开始作为服务的承载体出现，其中的一个具体事例就是无人驾驶的产品化。无人驾驶并不是一个单一的新技术，而是一系列技术的整合，通过众多技术的有效融合，在无人驾车的情况下安全地送达乘客。本章会介绍无人驾驶的分级、ADAS 中的关键应用、无人驾驶中涉及的多项技术并讨论如何安全高效地在无人驾驶系统中完成技术的整合。

1.1 正在走来的无人驾驶

预计到 2021 年，无人车将进入市场，从此开启一个崭新的阶段。^[1]世界经济论坛估计，汽车行业的数字化变革将创造 670 亿美元的价值，带来 3.1 万亿美元的社会效益，^[2]其中包括无人车的改进、乘客互联及整个交通行业生态系统的完善。

据估计，半自动驾驶和全自动驾驶汽车在未来几十年的市场潜力相当大。例如，到 2035 年，仅中国就将有约 860 万辆自动驾驶汽车，其中约 340 万辆为全自动无人驾驶，520 万辆为半自动驾驶。^[3]有行业主管部门人士认为，“中国轿车的销售，巴士、出租车和相关交通服务年收入有望超过 1.5 万亿美元。”波士顿咨询集团预测，“无人车的全球市场份额要达到 25%，需要花 15~20 年的时间。”由于无人车预计到 2021 年才上市，这意味着 2035—2040 年，无人车将占全球市场 25% 的份额。

无人驾驶之所以会给汽车行业带来如此大的变革，是因为无人车带来的影响是空前的。研究表明，在增强高速公路安全、缓解交通拥堵、减少空气污染等领域，无人驾驶会带来

颠覆性的改善。

1. 增强高速公路安全

高速公路事故是全世界面临的重大问题。在美国，每年估计有 35000 人死于车祸，在中国这一数字约为 260000 人。^[4] 日本每年高速公路事故死亡人数为 4000 左右。^[5] 根据世界卫生组织统计，全世界每年有 124 万人死于高速公路事故。^[6] 据估计，致命车祸每年会造成 2600 亿美元的损失，而车祸致伤会带来 3650 亿美元的损失。高速公路事故每年导致 6250 亿美元的损失。^[7] 美国兰德公司研究显示，“在 2011 年车祸死亡事故中 39% 涉及酒驾。”^[8] 几乎可以肯定，在这方面，无人车将带来大幅改善，避免车祸伤亡。在中国，约有 60% 的交通事故和骑车人、行人或电动自行车与小轿车和卡车相撞有关。^[9] 在美国的机动车事故中，有 94% 与人为失误有关。^[10] 美国高速公路安全保险研究所的一项研究表明，全部安装自动安全装置能使高速公路事故死亡数量减少 31%，每年将挽救 11000 条生命。^[11] 这类装置包括前部碰撞警告体系、碰撞制动、车道偏离警告和盲点探测。

2. 缓解交通拥堵

交通拥堵几乎是每个大都市都面临的问题。以美国为例，每位司机每年平均遇到 40 小时的交通堵塞，年均成本为 1210 亿美元。^[12] 在莫斯科、伊斯坦布尔、墨西哥城或里约热内卢，浪费的时间更长，“每位司机每年将在交通拥堵中度过超过 100 小时。”^[13] 在中国，汽车数量超过 100 万辆的城市有 35 个，超过 200 万辆的城市有 10 个。在最繁忙的市区，约有 75% 的道路会出现高峰拥堵。”中国私家车总数已达 1.26 亿辆，同比增加 15%，^[14] 仅北京就有 560 万辆汽车。^[15] Donald Shoup 的研究发现，都市区 30% 的交通拥堵是由于司机为了寻找附近的停车场而在商务区绕圈造成的。^[16] 这是交通拥挤、空气污染和环境恶化的重要原因。“在造成气候变化的二氧化碳排放中约有 30% 来自汽车”。^[17] 另外，根据估算，在都市中有 23%~45% 的交通拥堵中发生在道路交叉处。^[18] 交通灯和停车标志不能发挥作用，因为它们是静止的，无法将交通流量考虑其中。绿灯或红灯是按照固定间隔提前设定好的，不管某个方向的车流量有多大。一旦无人车逐渐投入使用，并占到车流量比较大的比例，车载感应器将能够与智能交通系统联合工作，优化道路交叉口的车流量。红绿灯的间隔也将是动态的，根据道路车流量实时变动。这样可以通过提高车辆通行效率，缓解拥堵。

3. 疏解停车难问题

完成停车时，无人车能将每侧人为预留的空间减少 10 厘米，每个停车位就可以减少 1.95 平方米，此外层高也可以按照车身进行设计。通过无人车与传统汽车使共享车库，所需要的车库空间将减少 26%。如果车库直供自动泊车汽车使用，则所需的车库空间将减少 62%。节省的土地可以用于建设其他对行车和行人更加友好的街道，同时也节省了消费者停车和取车的时间。

4. 减少空气污染

汽车是造成空气质量下降的主要原因之一。兰德公司的研究表明，“无人驾驶技术能提高燃料效率，通过更顺畅的加速、减速，能比手动驾驶提高 4%~10% 的燃料效率。”^[19] 由于工业区的烟雾与汽车数量有关，增加无人车的数量能减少空气污染。一项 2016 年的研究估计，“等红灯或交通拥堵时汽车造成的污染比车辆行驶时高 40%。”^[20] 无人车共享系统也能带来减排和节能的好处。德克萨斯大学奥斯汀分校的研究人员研究了二氧化硫、一氧化碳、氮氧化物、挥发性有机化合物、温室气体和细小颗粒物。结果发现，“使用无人车共享系统不仅节省能源，还能减少各种污染物的排放。”^[21] 约车公司 Uber 发现，该公司在旧金山和洛杉矶的车辆出行中分别有 50% 和 30% 是多乘客拼车。在全球范围内，这一数字为 20%。^[22] 无论是传统车，还是无人车，拼车的乘客越多，对环境越好，也越能缓解交通拥堵。改变一车一人的模式将能大大改善空气质量。

1.2 自动驾驶的分级

2013 年，美国国家公路交通安全管理局（NHTSA，制定各种监管和标准）发布了汽车自动化的五级标准，将自动驾驶功能分为 5 个级别：0~4 级^[23]，以应对汽车主动安全技术的爆发增长。先看 NHTSA 下的定义，如图 1-2 所示。

（1）Level 0：无自动化。没有任何自动驾驶功能、技术，司机对汽车所有功能拥有绝对控制权。驾驶员需要负责启动、制动、操作和观察道路状况。任何驾驶辅助技术，只要仍需要人控制汽车，都属于 Level 0。所以现有的前向碰撞预警、车道偏离预警，以及自动雨刷和自动前灯控制，虽然有一定的智能化，但是都仍属于 Level 0。

自动驾驶分级

美国国家公路交通安全管理局 (NHTSA)、美国汽车工程学会 (SAE) 自动驾驶分级标准							
分级	NHTSA	L0	L1	L2	L3	L4	L5
SAE	SAE	L0	L1	L2	L3	L4	L5
术语(SAE)	无自动化	驾驶支持	部分自动化	有条件自动化	高度自动化	完全自动化	
SAE定义	由人类驾驶员全权驾驶汽车，在行驶过程中可以得到警告	通过驾驶环境对方向盘和加速减速中的一项操作提供支持，其余由人类操作	通过驾驶环境对方向盘和加速减速中的多项操作提供支持，其余由人类操作	由无人驾驶系统完成所有的驾驶操作，根据系统要求，人类提供适当的应答	由无人驾驶系统完成所有的驾驶操作，根据系统要求，人类不一定提供所有的应答。限定道路和环境条件	由无人驾驶系统完成所有的驾驶操作，可能的情况下，人类接管，不限定道路和环境条件	
主体	驾驶员	人类驾驶员	人类驾驶员/系统	系统	系统	系统	系统
系统作用域	无						全域

图 1-2 NHTSA 和 SAE 对自动驾驶的分级比较

(2) Level 1：单一功能级的自动化。驾驶员仍然对行车安全负责，不过可以放弃部分控制权给系统管理，某些功能已经自动进行，比如常见的自适应巡航（Adaptive Cruise Control, ACC）、应急刹车辅助（Emergency Brake Assist, EBA）和车道保持（Lane-Keep Support, LKS）。Level 1 的特点是只有单一功能，驾驶员无法做到手和脚同时不操控。

(3) Level 2：部分自动化。司机和汽车来分享控制权，驾驶员在某些预设环境下可以不操作汽车，即手脚同时离开控制，但驾驶员仍需要随时待命，对驾驶安全负责，并随时准备在短时间内接管汽车驾驶权。比如结合了 ACC 和 LKS 形成的跟车功能。Level 2 的核心不在于要有两个以上的功能，而在于驾驶员可以不再作为主要操作者。Tesla 推送的 autopilot 也是 Level 2 的功能。

(4) Level 3：有条件自动化。在有限情况下实现自动控制，比如在预设的路段（如高速和人流较少的城市路段），汽车自动驾驶可以完全负责整个车辆的操控，但是当遇到紧急情况，驾驶员仍需要在某些时候接管汽车，但有足够的预警时间，如即将进入修路的路段（Road work ahead）。Level 3 将解放驾驶员，即对行车安全不再负责，不必监视道路状况。

(5) Level 4：完全自动化（无人驾驶），无须司机或乘客的干预。在无须人协助的情况下由出发地驶向目的地。仅需起点和终点信息，汽车将全程负责行车安全，并完全不依赖驾驶员干涉。行车时可以没有人乘坐（如空车货运）。

另一个对自动驾驶的分级来自美国机动工程师协会（SAE），其定义自动驾驶技术共

分为 0~5 级。^[24]SAE 的定义在自动驾驶 0~3 级与 NHTSA 一致,分别强调的是无自动化、驾驶支持、部分自动化与条件下的自动化。唯一的区别在于 SAE 对 NHTSA 的完全自动化进行了进一步细分,强调了行车对环境与道路的要求。SAE-Level4 下的自动驾驶需要在特定的道路条件下进行,比如封闭的园区或者固定的行车线路等,可以说是面向特定场景下的高度自动化驾驶。SAE-Level5 则对行车环境不加限制,可以自动地应对各种复杂的车辆、新人和道路环境。

综上所述,不同 Level 所实现的自动驾驶功能也是逐层递增的,ADAS (Advanced Driving Assistant System) 即高级驾驶辅助系统,属于自动驾驶 0~2 级。如表 1-1 所示,L0 中实现的功能仅能够进行传感探测和决策报警,比如夜视系统、交通标识识别、行人检测、车道偏离警告等。L1 实现单一控制类功能,如支持主动紧急制动、自适应巡航控制系统等,只要实现其中之一就可达到 L1。L2 实现了多种控制类功能,如具有 AEB 和 LKA 等功能的车辆。L3 实现了特定条件下的自动驾驶,当超出特定条件时将由人类驾驶员接管驾驶。SAE 中的 L4 是指在特定条件下的无人驾驶,如封闭园区固定线路的无人驾驶等,例如百度在乌镇景区运营的无人驾驶服务。而 SAE 中的 L5 就是终极目标,完全无人驾驶。无人驾驶就是自动驾驶的最高级,它是自动驾驶的最终形态。

表 1-1 逐层递增的自动驾驶功能

NHTSA	L0	L1	L2	L3	L4	
SAE	L0	L1	L2	L3	L4	L5
	无自动化	驾驶支持	部分自动化	有条件自动化	高度自动化	完全自动化
功能	夜视 行人检测 交通标志识别 盲点检测 并线辅助 后排路口交通警报 车道偏离警告	自适应巡航驾驶系统 自动紧急制动 停车辅助系统 前向碰撞预警系统 车身电子稳定系统	车道保持辅助系统	拥挤辅助驾驶	停车场自动泊车	
特征	传感探测和决策警报	单一功能（以上之一）	组合功能（L1/L2 组合）	特定条件 部分任务	特定条件 全部任务	全部条件 全部任务
ADAS				自动驾驶		

全自动无人车可能比半自动驾驶汽车更安全,因为其可以在车辆行驶时排除人为错误和不明智的判断。例如,弗吉尼亚理工大学交通学院的调查表明,“L3 级自动驾驶车辆的司机回应接管车辆的请求平均需要 17 秒,而在这个时间内,一辆时速 65 英里(105 千米)的汽车已经开出 1621 英尺(494 米)——超过 5 个足球场的长度。”百度的工程师也发现了类似的结果。司机从看到路面物体到踩刹车需要 1.2 秒,远远长于车载计算机所用的 0.2 秒。这一时间差意味着,如果汽车时速是 120 千米(75 英里),等到司机停车时,车子已经开出了 40 米(44 码),而如果是车载计算机做判断,则开出的距离只有 6.7 米(7 码)。在很多事故中,这一差距将决定乘客的生死。由此可见,站在自动驾驶最高级的无人驾驶才是汽车行业未来发展的“终极目标”。

1.3 无人驾驶系统简介

无人驾驶系统是一个复杂的系统,如图 1-3 所示,系统主要由三部分组成:算法端、Client 端和云端。其中算法端包括面向传感、感知和决策等关键步骤的算法;Client 端包括机器人操作系统及硬件平台;云端包括数据存储、模拟、高精度地图绘制及深度学习模型训练。

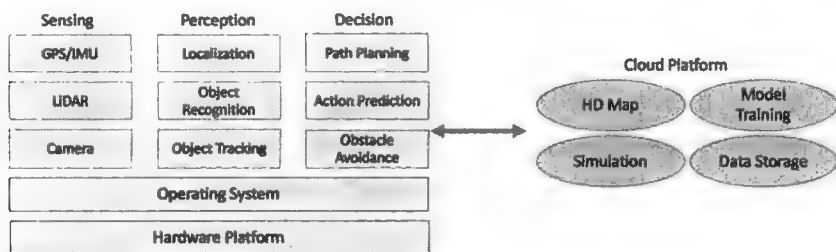


图 1-3 无人驾驶系统架构图

算法端从传感器原始数据中提取有意义的信息以了解周遭的环境情况,并根据环境变化做出决策。Client 端融合多种算法以满足实时性与可靠性的要求。举例来说,传感器以 60Hz 的速度产生原始数据,Client 端需要保证最长的流水线处理周期也能在 16ms 内完成。云平台为无人车提供离线计算及存储功能。通过云平台,我们能够测试新的算法、更新高精度地图并训练更加有效的识别、追踪和决策模型。

1.3.1 无人驾驶算法

算法系统由几部分组成：第一，传感，并从传感器原始数据中提取有意义信息；第二，感知，以定位无人车所在位置及感知现在所处的环境；第三，决策，以便可靠、安全地抵达目的地。

1. 传感

通常来说，一辆无人车装备有许多不同类型的主传感器。每一种类型的传感器各自有不同的优劣，因此，来自不同传感器的传感数据应该有效地进行融合。现在无人驾驶中普遍使用的传感器包括以下几种。

(1) GPS/IMU：GPS/IMU 传感系统通过高达 200 Hz 频率的全球定位和惯性更新数据，以帮助无人车完成自我定位。GPS 是一个相对准确的定位用传感器，但是它的更新频率过低，仅有 10Hz，不足以提供足够实时的位置更新。IMU 的准确度随着时间降低，因此在长时间距离内并不能保证位置更新的准确性；但是，它有着 GPS 所欠缺的实时性，IMU 的更新频率可以达到 200Hz 或者更高。通过整合 GPS 与 IMU，我们可以为车辆定位提供既准确又足够实时的位置更新。

(2) LIDAR：激光雷达可被用来绘制地图、定位及避障。雷达的准确率非常高，因此在无人车设计中雷达通常被作为主传感器使用。激光雷达是以激光为光源，通过探测激光与被探测物相互作用的光波信号来完成遥感测量。激光雷达可以用来产生高精度地图，并针对高精地图完成移动车辆的定位，以及满足避障的要求。以 Velodyne 64-束激光雷达为例，它可以完成 10Hz 旋转并且每秒可达到 130 万次读数。

(3) 摄像头：摄像头被广泛使用在物体识别及物体追踪等场景中，在车道线检测、交通灯侦测、人行道检测中都以摄像头为主要解决方案。为了加强安全性，现有的无人车实现通常在车身周围使用至少八个摄像头，分别从前、后、左、右四个维度完成物体发现、识别、追踪等任务。这些摄像头通常以 60Hz 的频率工作，当多个摄像头同时工作时，将产生高达 1.8GB 每秒的巨额数据量。

(4) 雷达和声呐：雷达把电磁波的能量发射至空间中某一方向，处在此方向上的物体反射该电磁波，雷达通过接收此反射波，以提取该物体的某些有关信息，包括目标物体至雷达的距离、距离变化率或径向速度、方位、高度等。雷达和声呐系统是避障的最后一道保障。雷达和声呐产生的数据用来表示在车的前进方向上最近障碍物的距离。一旦系统检

测到前方不远有障碍物出现，则有极大的相撞危险，无人车会启动紧急刹车以完成避障。因此，雷达和声呐系统产生的数据不需要过多的处理，通常可直接被控制处理器采用，并不需要主计算流水线的介入，因此可实现转向、刹车或预张紧安全带等紧急功能。

2. 感知

在获得传感信息之后，数据将被推送至感知子系统以充分了解无人车所处的周遭环境。在这里感知子系统主要做的是三件事：定位、物体识别与追踪。

1) 定位

GPS 以较低的更新频率提供相对准确的位置信息，IMU 则以较高的更新频率提供准确性偏低的位置信息。我们可以使用卡尔曼滤波整合两类数据各自的优势，合并提供准确且实时的位置信息更新。如图 1-4 所示，IMU 每 5ms 更新一次，但是期间误差不断累积精度不断降低。所幸的是，每 100ms，我们可以得到一次 GPS 数据更新，以帮助我们校正 IMU 积累的误差。因此，我们最终可以获得实时并准确的位置信息。然而，我们不能仅仅依靠这样的数据组合完成定位工作。原因有三：其一，这样的定位精度仅在一米之内；其二，GPS 信号有着天然的多路径问题将引入噪声干扰；其三，GPS 必须在非封闭的环境下工作，因此在诸如隧道等场景中 GPS 都不适用。

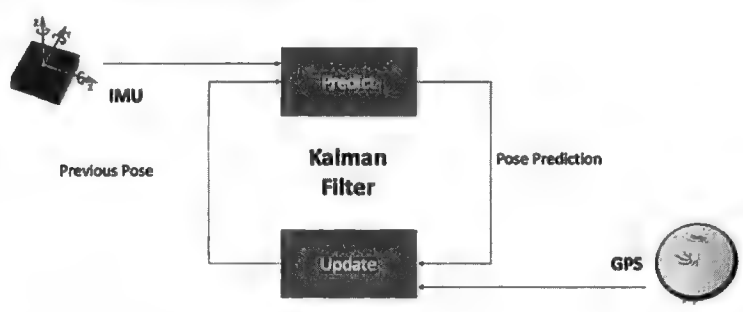


图 1-4 基于 GPS/IMU 定位的原理图

因此作为补充方案，摄像头也被用于定位。简化来说，如图 1-5 所示，基于视觉的定位由三个基本步骤组成：① 通过对立体图像的三角剖分，首先获得视差图用以计算每个点的深度信息；② 通过匹配连续立体图像帧之间的显著特征，可以通过不同帧之间的特征建立相关性，并由此估计这两帧之间的运动情况；③ 通过比较捕捉到的显著特征和已知地图上的点计算车辆的当前位置。然而，基于视觉的定位方法对照明条件非常敏感，因

此其使用受限且可靠性有限。

因此，借助于大量粒子滤波的激光雷达通常被用作车辆定位的主传感器。由激光雷达产生的点云对环境进行了“形状化描述”，但并不足以区分各自不同的点。通过粒子滤波，系统可将已知地图与观测到的具体形状进行比较以减少位置的不确定性。

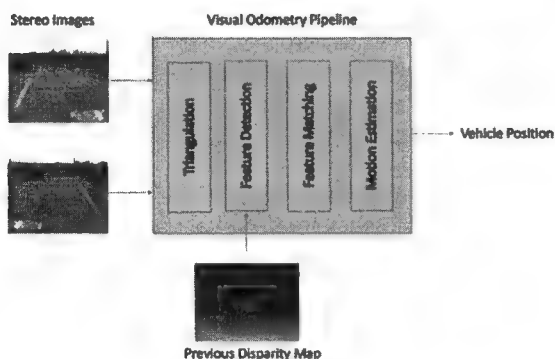


图 1-5 基于立体视觉测距的流程图

为了在地图中定位运动的车辆，可以使用粒子滤波的方法关联已知地图和激光雷达测量过程。粒子滤波可以在 10cm 的精度内达到实时定位的效果，在城市的复杂环境中尤为有效。然而，激光雷达也有其固有的缺点：如果空气中有悬浮的颗粒（比如雨滴或者灰尘），那么测量结果将受到极大的扰动。因此，如图 1-6 所示，我们需要利用多种传感器融合技术进行多类型传感数据融合，处理以整合所有传感器的优点，完成可靠并精准的定位。

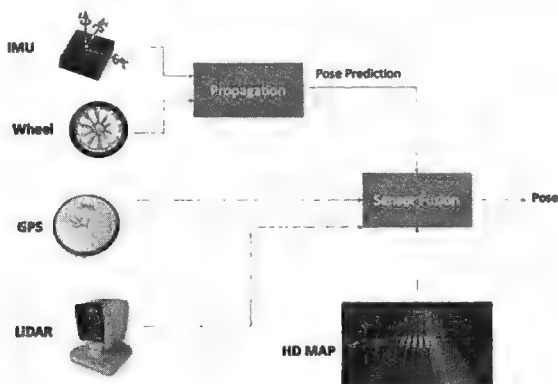


图 1-6 定位中的多传感器融合

2) 物体识别与跟踪

激光雷达可提供精准的深度信息,因此常被用于在无人驾驶中执行物体识别和跟踪的任务。近年来,深度学习技术得到了快速的发展,通过深度学习可达到较显著的物体识别和跟踪精度。

卷积神经网络(CNN)是一类在物体识别中被广泛应用的深度神经网络。通常,CNN由三个阶段组成:① 卷积层使用不同的滤波器从输入图像中提取不同的特征,并且每个过滤器在完成训练阶段后都将抽取出一套“可供学习”的参数;② 激活层决定是否启动目标神经元;③ 汇聚层压缩特征映射图所占用的空间以减少参数的数目,并由此降低所需的计算量;④ 一旦某物体被CNN识别出来,下一步将自动预测它的运行轨迹或进行物体跟踪,如图1-7所示。



图 1-7 物体识别和跟踪示意

物体跟踪可以被用来跟踪邻近行驶的车辆或者路上的行人,以保证无人车在驾驶的过程中不会与其他移动的物体发生碰撞。近年来,相比传统的计算机视觉技术,深度学习技术已经展露出极大的优势,通过使用辅助的自然图像,离线的训练过程可以从中学习图像的共有属性以避免视点及车辆位置变化造成的偏移,离线训练好的模型直接应用在在线的物体跟踪中。

3. 决策

在决策阶段,行为预测、路径规划及避障机制三者结合起来实时地完成无人驾驶动作规划。

1) 行为预测

在车辆驾驶中主要考验的是司机如何应对其他行驶车辆的可能行为,这种预判断直接影响司机本人的驾驶决策,特别是在多车道环境或者交通灯变灯的情况下,司机的预测决定了下一秒行车的安全。因此,过渡到无人驾驶系统中,决策模块如何根据周围车辆的行驶状况决策下一秒的行驶行为显得至关重要。

为了预测其他车辆的行驶行为,可以使用随机模型产生这些车辆的可达位置集合,并采用概率分布的方法预测每一个可达位置集的相关概率,如图 1-8 所示。

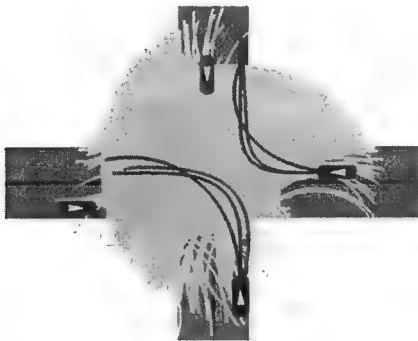


图 1-8 面向行为预测的随机模型示意

2) 路径规划

为无人驾驶在动态环境中进行路径规划是一件非常复杂的事情,尤其是在车辆全速行驶的过程中,不当的路径规划有可能造成致命的伤害。路径规划中采取的一个方法是使用完全确定模型,它搜索所有可能的路径并利用代价函数的方式确定最佳路径。然后,完全确定模型对计算性能有着非常高的要求,因此很难在导航过程中达到实时的效果。为了避免计算复杂性并提供实时的路径规划,使用概率性模型成为了主要的优化方向。

3) 避障

安全性是无人驾驶中最重要的考量,我们将使用至少两层级的避障机制来保证车辆不会在行驶过程中与障碍物发生碰撞。第一层级是基于交通情况预测的前瞻层级。交通情况预测机制根据现有的交通状况如拥堵、车速等,估计出碰撞发生时间与最短预测距离等参数。基于这些估计,避障机制将被启动以执行本地路径重规划。如果前瞻层级预测失效,则第二级实时反应层将使用雷达数据再次进行本地路径重规划。一旦雷达侦测到路径前方

出现障碍物，则立即执行避障操作。

1.3.2 用户端系统

用户端系统整合上述避障、路径规划等算法，以满足可靠性及实时性等要求。用户端系统需要克服三个方面的问题：其一，系统必须确保捕捉到的大量传感器数据可以及时快速地得到处理；其二，如果系统的某部分失效，则系统需要有足够的健壮性能从错误中恢复；其三，系统必须在设计的能耗和资源限定下有效地完成所有的计算操作。

1. 机器人操作系统

机器人操作系统 ROS 是现如今广泛被使用的、专为机器人应用裁剪的、强大的分布式计算框架。ROS 为机器人应用提供诸如硬件抽象描述、底层驱动程序管理、消息管理与传递、程序发行包管理等基本功能，同时也提供一系列工具和库用于开发、获取和运行机器人应用。节点（node）是 ROS 中的基本单位，其粒度范围很广，小到一个传感器大到一个完整的机器人都可以是一个节点。每一个机器人任务，比如避障，也作为 ROS 中的一个节点存在。节点与节点之间通过消息互相通信，其通信是端对端的，消息可以按照主题分类，也可以包装成远程服务调用的形式。ROS 中的节点管理器和消息管理器提供命名和查找服务以方便节点在运行时能找到彼此，如图 1-9 所示。

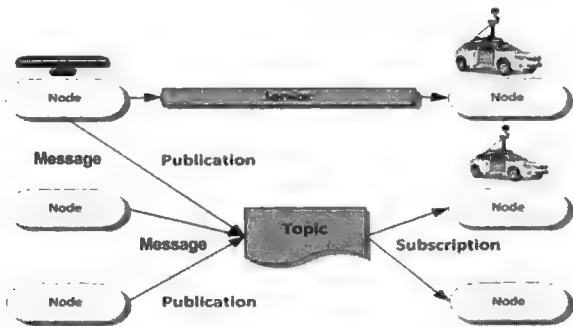


图 1-9 ROS 操作系统结构示意图

ROS 非常适用于无人驾驶的场景，但是仍有一些问题需要解决。

- 可靠性：ROS 使用单主节点结构，并且没有监控机制以恢复失效的节点。
- 性能：当节点之间使用广播消息的方式通信时，将产生多次信息复制导致性能下降。
- 安全：ROS 中没有授权和加密机制，因此安全性受到很大的威胁。

尽管 ROS 2.0 承诺将解决上述问题，但是现有的 ROS 版本中仍然没有相关的解决方案。因此，为了在无人驾驶中使用 ROS，我们需要自行克服这些难题。

1) 可靠性

现有的 ROS 实现只有一个主节点，因此当主节点失效时，整个系统也随之崩溃。这对行驶中的汽车而言是致命的缺陷。为了解决此问题，我们在 ROS 中使用类似于 ZooKeeper 的方法。如图 1-10 所示，改进后的 ROS 结构包括一个关键主节点及一个备用主节点。如果关键主节点失效，则备用主节点将被自动启用以确保系统能够无缝地继续运行。此外，ZooKeeper 机制将监控并自动重启失效节点，以确保整个 ROS 系统在任何时刻都是双备份模式。

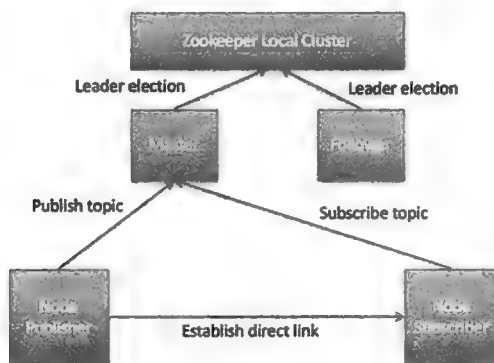


图 1-10 面向 ROS 的 ZooKeeper 结构

2) 性能

性能是现有 ROS 版本中有欠考虑的部分，ROS 节点之间的通信非常频繁，因此设计高效的通信机制对保证 ROS 的性能势在必行。首先，本地节点在与其他节点通信时使用回环机制，并且每一次回环通信的执行都将完整地通过 TCP/IP 全协议栈，从而引入高达 20 微秒的时延。为了消除本地节点通信的代价，我们不再使用 TCP/IP 的通信模式，取而代之地采用共享内存的方法完成节点通信。其次，当 ROS 节点广播通信消息时，消息被多次复制与传输，消耗了大量的系统带宽。如果改成目的地更明确的多路径传输机制则将极大地改善系统的带宽与吞吐量，如图 1-11 所示。

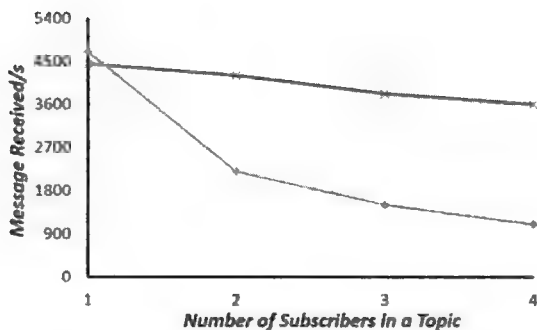


图 1-11 多路传播和广播的通信性能比较

3) 安全

安全是 ROS 系统中最重要需求。如果一个 ROS 节点被挟制后，则会不停地进行内存分配，整个系统最终将因内存耗尽而导致剩余节点失效继而全线崩溃。在另一个场景中，因为 ROS 节点本身没有加密机制，黑客可以很容易地在节点之间窃听消息并完成系统入侵。

为了解决安全问题，我们使用 Linux containers (LXC) 的方法限制每一个节点可供使用的资源数，并采用沙盒的方式以确保节点的运行独立，这样以来可最大限度地防止资源泄露。同时，我们为通信消息进行了加密操作，以防止其被黑客窃听。

2. 硬件平台

为了深入理解设计无人驾驶硬件平台中可能遇到的挑战，让我们来看看现有的领先无人车驾驶产品的计算平台构成。此平台由两个计算盒组成，每一个装备有 INTEL Xeon E5 处理器及 4 到 8 个 Nvidia Tesla K80 GPU 加速器。两个计算盒执行完全一样的工作，第二个计算盒作为计算备份以提高整个系统的可靠性，一旦第一个计算盒发生故障，计算盒二可以无缝地接手所有的计算工作。

在极端的情况下，如果两个计算盒都在峰值下运行，及时功耗将高达 5000W，同时也将遭遇非常严重的发热问题。因此，计算盒必须配备有额外的散热装置，可采用多风扇或者水冷的方案。同时，每一个计算盒的造价非常昂贵，高达 2 万至 3 万美元，致使现有无人车方案对普通消费者而言无法承受。

现有无人车设计方案中存在的功耗问题、散热问题及造价问题使得无人驾驶进入普罗大众显得遥不可及。为了探索无人驾驶系统在资源受限及能耗受限时运行的可行性，我们在 ARM 面向移动市场的 SoC 实现了一个简化的无人驾驶系统，实验显示，在峰值情况下能耗仅为 15W。

非常惊人地，在移动类 SoC 上，无人驾驶系统的性能反而带给了我们一些惊喜：定位算法可以达到每秒 25 帧的处理速度，同时能维持图像生成的速度在 30 帧每秒。深度学习则能在一秒内完成 2~3 个物体的识别工作。路径规划和控制则可以在 6 毫秒之内完成规划工作。在性能的驱动下，我们可以在不损失任何位置信息的情况下达到每小时 5 迈的行驶速度。

1.3.3 云平台

无人车是移动系统，因此需要云平台的支持。云平台主要从分布式计算及分布式存储两方面对无人驾驶系统提供支持。无人驾驶系统中很多的应用，包括用于验证新算法的仿真应用、高精度地图产生和深度学习模型训练都需要云平台的支持。我们使用 Spark 构建了分布式计算平台，使用 OpenCL 构建了异构计算平台，使用 Alluxio 作为内存存储平台。通过这三个平台的整合，可以为无人驾驶提供高可靠、低延迟及高吞吐的云端支持。

1. 仿真

当我们为无人驾驶开发出新算法时，需要先通过仿真对此算法进行全面测试，测试通过之后才进入真车测试环节。真车测试的成本非常高昂并且迭代周期异常漫长，因此仿真测试的全面性和正确性对降低生产成本和生产周期尤为重要。在仿真测试环节，我们通过在 ROS 节点回放真实采集的道路交通情况，模拟真实的驾驶场景，完成对算法的测试。如果没有云平台的帮助，单机系统耗费数小时才能完成一个场景下的模拟测试，既耗时测试覆盖面又有限。

在云平台中，Spark 管理着分布式的多个计算节点，在每一个计算节点中，都可以部署一个场景下的 ROS 回访模拟。在无人驾驶物体识别测试中，单服务器需耗时 3 小时完成算法测试，如果使用 8 机 Spark 机群，则时间可以缩短至 25 分钟，如图 1-12 所示。

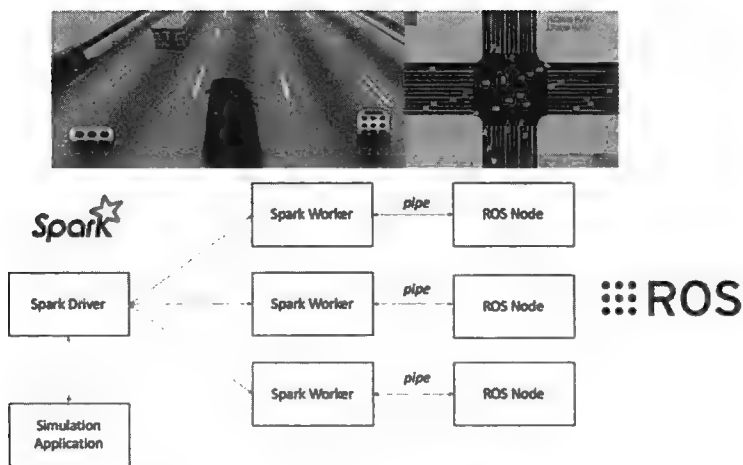


图 1-12 基于 Spark 和 ROS 的模拟平台架构

2. 高精度地图生成

如图 1-13 所示，高精度地图的产生过程非常复杂，涉及原始数据处理、点云生成、点云对齐、2D 反射地图生成、高精地图标注、地图生成等阶段。使用 Spark 可以将所有这些阶段整合成一个 Spark 作业。由于 Spark 天然的内存计算的特性，在作业运行过程中产生的中间数据都存储在内存中。当整个地图生产作业提交之后，不同阶段之间产生的大量数据不需要使用磁盘存储，数据访问速度加快，从而极大提高了高精度地图产生的性能。

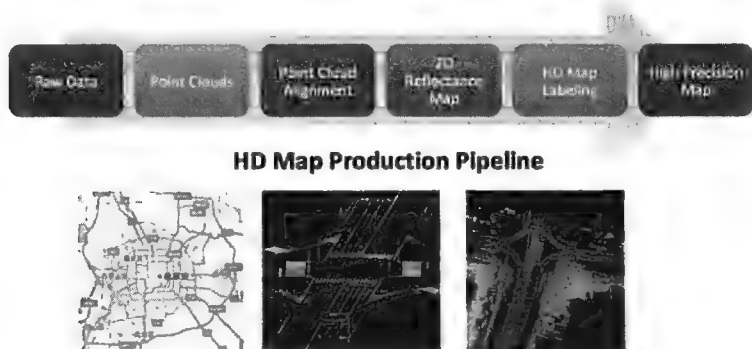


图 1-13 基于云平台的高精度地图生成流程图

1.4 序幕刚启

无人驾驶作为人工智能的一个重大应用发现从来就不是某一项单一的技术,它是众多技术的整合。它需要有算法上的创新、系统上的融合,以及来自云平台的支持。无人驾驶序幕刚启,其中有着千千万万的机会亟待发掘。在此背景之下,过去的几年中,自动驾驶产业化在多个方面取得了很大进步,其中合作共享已成为共识,产业链不断整合,业界企业相继开展合作,传感器价格将不断下降,预计在 2020 年,将有真正意义上的无人车面市,让我们拭目以待。

1.5 参考资料

- [1] 2016 年 4 月 8 日 Doug Newcomb 载于《福布斯》的文章“Volvo’s China 100-Vehicle Autonomous Car Trial Pushes Self-Driving Technology, Regulation”; 2016 年 7 月 4 日《纽约时报》上 John Markoff 的文章“Tesla and Google Take Different Roads to Self-Driving Car”。
- [2] 2016 年 1 月,世界经济论坛与埃森哲的合作研究“Digital Transformation of Industries: Automotive Industry”,作者 Bruce Weindelt。
- [3] 2016 年 6 月 22 日《中国日报》文章“Officials Want to Open Way for Autonomous Driving”。
- [4] 2016 年 5 月 30 日《纽约时报》文章“Beijing’s Electric Bikes, the Wheels of E-Commerce, Face Traffic Backlash”,作者 Chris Buckley。
- [5] 2015 年 10 月 28 日彭博技术文章“Japan’s Carmakers Proceed With Caution on Self-Driving Cars”,作者 Jie Ma。
- [6] 世界卫生组织 2010 年报告“Global Health Observatory Data: Number of Road Traffic Deaths”。
- [7] 2014 年 2 月 25 日摩根斯坦利研究“Nikola’s Revenge: TSLA’s New Path of Disruption”,第 24~26 页。
- [8] 兰德公司 2016 年报告“Autonomous Vehicle Technology: A Guide for

- Policymakers”, 作者 James Anderson、Nidhi Kalra、Karlyn Stanley、Paul Sorensen、Constantine Samaras 和 Oluwatobi Oluwatola.
- [9] 《纽约时报》2016 年 5 月 30 日文章 “Beijing’s Electric Bikes, the Wheels of E-Commerce, Face Traffic Backlash”, 作者 Chris Buckley.
- [10] 2015 年 11 月 18 日 Nathaniel Beuse 在众议院监督和政府改革委员会的证词。2016 年 6 月 2 日《华尔街日报》文章 “Baidu Plans to Mass Produce Autonomous Cars in Five Years”, 作者 Alyssa Abkowitz.
- [11] Delphi 公司的 Glen De Vos 于 2016 年 3 月 15 在参议院商业、科学和技术委员会听证会上的证词, 第 4 页.
- [12] 美国交通部 2015 年报告 “Beyond Traffic, 2045: Trends and Choices”, 第 11 页.
- [13] 2016 年 1 月世界经济论坛与埃森哲联合报告 “Digital Transformation of Industries: Automotive Industry”, 第 4 页, 作者 Bruce Weindelt.
- [14] 2016 年 4 月 21 日李书福《华尔街日报》文章 “Paving the Way for Autonomous Cars in China” .
- [15] 《纽约时报》2016 年 5 月 30 日文章 “Beijing’s Electric Bikes, the Wheels of E-Commerce, Face Traffic Backlash”, 作者 Chris Buckley.
- [16] Access 2007 年第 30 卷文章 “Cruising for Parking”, 第 16~22 页, 作者 Daniel Shoup.
- [17] 2016 年 1 月世界经济论坛与埃森哲联合报告 “Digital Transformation of Industries: Automotive Industry”, 第 4 页, 作者 Bruce Weindelt.
- [18] 2016 年 7 月 12 日采访百度专家.
- [19] 兰德公司 2016 年报告 “Autonomous Vehicle Technology: A Guide for Policymakers”, 作者 James Anderson、Nidhi Kalra、Karlyn Stanley、Paul Sorensen、Constantine Samaras 和 Oluwatobi Oluwatola, 第 xvi 页.
- [20] 2016 年 8 月 29 日《纽约时报》文章 “Stuck in Traffic, Polluting the Inside of Our Cars”, 作者 Tatiana Schlossberg.

- [21] Daniel Fagnant 和 Kara Kockelman 在 2014 年 1 月交通研究理事会第 93 次年会提交的文章 “The Travel and Environmental Implications of Shared Autonomous Vehicles Using Agent-Based Model Scenarios”, 第 1~13 页.
- [22] 2016 年 6 月 30 日采访 Uber 公司 Ashwini Chabra.
- [23] SAE Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles: http://standards.sae.org/j3016_201609/.
- [24] National Highway Traffic Safety Administration Preliminary Statement of Policy Concerning Automated Vehicles.
- [25] Creating Autonomous Vehicle Systems: <https://www.oreilly.com/ideas/creating-autonomous-vehicle-systems>.

2

光学雷达在无人驾驶技术中的应用

无人车的成功涉及了包括高精地图、实时定位，以及障碍物检测等多个技术，而这些技术都离不开光学雷达。本章简单介绍了无人驾驶技术，并且深入解释了光学雷达如何被广泛应用到无人车的各个技术中。本章首先介绍光学雷达的工作原理包括如何通过激光扫描出点云。接下来详细解释光学雷达在无人驾驶技术中的应用，包括地图绘制、定位，以及障碍物检测。最后讨论光学雷达技术目前面临的挑战，包括外部环境干扰、数据量大、成本高等问题。

2.1 无人驾驶技术简介

无人驾驶技术是多个技术的集成，包括传感器、定位与深度学习、高精地图、路径规划、障碍物检测与规避、机械控制、系统集成与优化、能耗与散热管理等。图 2-1 所示为无人车的通用系统图，虽然现有的多种无人车在实现上有许多不同，但是在系统架构上都大同小异。在系统的感知端由不同的传感器组成，其中包括 GPS 用于定位，光学雷达用于定位及障碍物检测，照相机用于深度学习的物体识别，以及定位辅助。

在传感器信息采集后，进入了感知阶段，主要是定位与物体识别。在这个阶段，可以

用数学的方法,比如 Kalman Filter 与 Particle Filter 等算法,对各种传感器信息进行融合,并得出当前最大概率的位置。如果使用光学雷达为主要的定位传感器,则可以通过光学雷达扫描回来的信息跟已知的高精地图做对比,从而得出当前的车辆位置。如果当前没有地图,那么甚至可以把当前的光学雷达扫描与之前的光学雷达扫描用 ICP 算法做对比,从而推算出当前的车辆位置。在得出基于光学雷达的位置预测后,可以用数学的方法与其他的传感器信息进行融合,推算出更精准的位置信息。

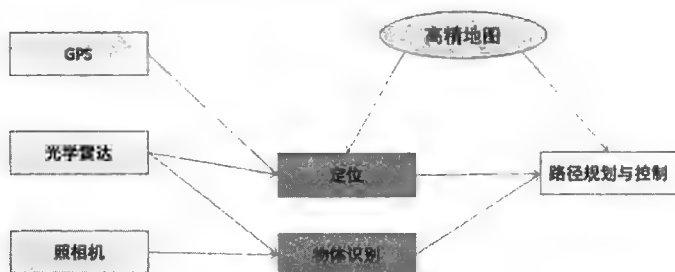


图 2-1 无人车通用系统架构示意图

最后,我们进入了计划与控制阶段。在这个阶段,我们根据位置信息及识别出的图像信息(比如红绿灯),实时调节车辆的行车计划,并把行车计划转化成控制信号去操控车辆。全局的路径规划可以用 A-Star 类似的算法实现,本地的路径规划可以用 DWA 等算法实现。

2.2 光学雷达基础知识

无人车涉及了包括高精地图、实时定位,以及障碍物检测等多个技术,而这些技术都离不开光学雷达。本节简单介绍光学雷达的工作原理,特别是产生点云的过程^[1]。

2.2.1 工作原理

光学雷达(Light Detection And Ranging, LiDAR)是一种光学遥感技术,它通过首先向目标物体发射一束激光,然后根据接收—反射的时间间隔确定目标物体的实际距离。然后根据距离及激光发射的角度,通过简单的几何变化推导出物体的位置信息。由于激光的传播受外界影响小, LiDAR 能够检测的距离一般可达 100m 以上。与传统雷达使用无线电波相比较, LiDAR 使用激光射线,商用 LiDAR 使用的激光射线波长一般在 600~1000nm,

远远低于传统雷达所使用的波长。因此，LiDAR 在测量物体距离和表面形状上可达到更高的精准，一般精准度可以达到厘米级。

LiDAR 系统一般分为三个部分：第一部分是激光发射器，发射出波长为 600~1000nm 的激光射线；第二部分是扫描与光学部件，主要用于收集反射点距离与该点发生的时间和水平角度（Azimuth）；第三部分是感光部件，主要检测返回光的强度。因此，我们检测到的每一个点都包括了空间坐标信息（ x, y, z ）及光强度信息*i*。光强度与物体的光反射度（reflectivity）直接相关，所以从检测到的光强度我们也可以对检测到的物体有初步判断。

2.2.2 什么是点云

无人车所使用的 LiDAR 并不是静止不动的。在无人车行驶的过程中，LiDAR 同时以一定的角速度匀速转动，在这个过程中不断地发出激光并收集反射点的信息，以便得到全方位的环境信息。LiDAR 在收集反射点距离的过程中也会同时记录下该点发生的时间和水平角度（Azimuth），并且每个激光发射器都有其编号和固定的垂直角度，根据这些数据就可以计算出所有反射点的坐标。LiDAR 每旋转一周，收集到的所有反射点坐标的集合就形成了点云（Point Cloud）。

如图 2-2 所示，LiDAR 通过激光反射可以测出和物体的距离（distance），因为激光的垂直角度是固定的，记作 a ，这里我们可以直接求出 z 轴坐标为 $\sin(a) \cdot \text{distance}$ 。由 $\cos(a) \cdot \text{distance}$ 可以得到 distance 在 xy 平面的投影，记作 xy_dist 。LiDAR 在记录反射点距离的同时也会记录下当前 LiDAR 转动的水平角度 b ，这样根据简单的集合转换就可以得到该点的 x, y 坐标分别为 $\cos(b) \cdot xy_dist$ 和 $\sin(b) \cdot xy_dist$ 。

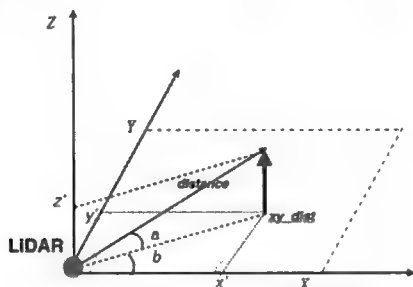


图 2-2 点云产生的坐标示意图

2.3 LiDAR 在无人驾驶技术中的应用领域

本节介绍光学雷达是如何应用在无人驾驶技术中的，特别是面向高精地图的绘制、基于点云的定位，以及障碍物检测。

2.3.1 高清地图的绘制

这里的高清地图不同于我们日常用到的导航地图。高清地图是由众多的点云拼接而成，其主要用于无人车的精准定位。高清地图的绘制也是通过 LiDAR 完成的。安装 LiDAR 的地图数据采集车在想要绘制高清地图的路线上多次反复行驶并收集点云数据。后期会经过人工标注，首先将过滤一些点云图中的错误信息，例如由路上行驶的汽车和行人反射所形成的点，然后对多次收集到的点云进行对齐拼接形成最终的高清地图。^{[2][3]}

2.3.2 基于点云的定位

首先讲讲定位的重要性。很多人都有这样的疑问：如果有了精准的 GPS，就知道了当前的位置，还需要定位吗？其实不然。目前高精度的军用差分 GPS 在静态时确实可以在“理想”的环境下达到厘米级的精度。这里的“理想”环境是指大气中没有过多的悬浮介质而且测量时 GPS 有较强的接收信号。然而，无人车是在复杂的动态环境中行驶，尤其在大城市中，由于各种高大建筑物的阻拦。GPS 多路径反射（Multi-Path）的问题会更明显。这样得到的 GPS 定位信息很容易就有几十厘米甚至几米的误差。对于在有限宽度上高速行驶的汽车，这样的误差很有可能导致交通事故。因此，必须要有 GPS 之外的手段增强无人车定位的精度。

上面提到过，LiDAR 会在车辆行驶的过程中不断地收集点云来了解周围的环境。我们可以很自然地想到利用这些观察到的环境信息帮助我们定位^{[4][5][6][7][8]}。可以把这个问题用下面这个简化的概率问题表示：已知 t_0 时刻的 GPS 信息， t_0 时刻的点云信息，以及无人车 t_1 时刻可能在的三个位置： P_1 、 P_2 和 P_3 （这里为了简化问题，假设无人车会在这三个位置中的某一个）。求 t_1 时刻车在这三点的概率。根据贝叶斯法则，无人车的定位问题可以简化为下面这个概率公式：

$$P(X_t) \approx P(Z_t|X_t) \cdot \overline{P(X_t)}$$

右侧第一项 $P(Z_t|X_t)$ 表示给定当前位置，观测到点云信息的概率分布。其计算方式一般分为局部估计和全局估计两种。局部估计较简单的做法就是通过当前时刻点云和上一时

刻点云的匹配,借助几何上的推导,估计出无人车在当前位置的可能性。全局估计就是利用当前时刻的点云和上面提到过的高清地图做匹配,可以得到当前车相对地图上某一位置的可能性。在实际中一般会将两种定位方法结合使用。第二项 $\overline{P(X_t)}$ 表示对当前位置的预测的概率分布,这里可以简单地用GPS给出的位置信息作为预测。通过计算 P_1 、 P_2 和 P_3 这三个点的后验概率,可以估算出无人车在哪一个位置的可能性最高。通过对两个概率分布相乘,可以很大程度上提高无人车定位的准确度,如图2-3所示。

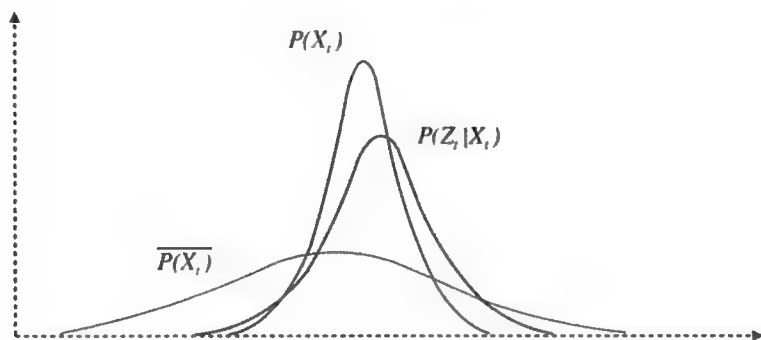


图 2-3 基于点云的定位

2.3.3 障碍物检测

众所周知,在机器视觉中一个比较难解决的问题就是判断物体的远近,基于单一摄像头抓取的2D图像无法得到准确的距离信息,而基于多摄像头生成深度图的方法又需要很大的计算量,不能很好地满足无人车在实时性上的要求。另一个棘手的问题是光学摄像头受光照条件的影响巨大。物体的识别准确度很不稳定。图2-4所示为由于光线条件不好,导致图像特征匹配的问题:由于照相机曝光不充分,左侧图中的特征点在右侧图中没有匹配成功。图2-5中左侧图展示了2D物体特征匹配成功的例子:啤酒瓶的模板可以在2D图像中被成功地识别出来,但是如果将镜头拉远,如图2-5中右图所示,则只能识别出右侧的啤酒瓶是附着在另一个3D物体的表面而已。2D的物体识别由于维度缺失的问题很难在这个情境下做出正确的识别。



图 2-4 暗光条件下图像特征匹配的挑战



图 2-5 2D 图像识别中存在的问题

而利用 LiDAR 生成的点云可以很大程度上解决上述两个问题,借助 LiDAR 本身的特性可以对反射障碍物的远近、高低,甚至是表面形状有较准确的估计,从而大大提高障碍物检测的准确度,而且在算法的复杂度上低于基于摄像头的视觉算法,因此更能满足无人车的实时性需求。

2.4 LiDAR 技术面临的挑战

前文我们专注于 LiDAR 对无人驾驶系统的帮助,但是在实际应用中, LiDAR 也面临着许多挑战。要想把无人车系统产品化,必须解决这些问题。本节讨论 LiDAR 的技术挑战、计算性能挑战,以及价格挑战。

2.4.1 技术挑战：空气中的悬浮物

LiDAR 的精度也会受到天气的影响。空气中悬浮物会对光速产生影响。大雾及雨天都会影响 LiDAR 的精度。测试环境为小雨的降雨量小于 10mm/h, 中雨的降雨量在 10mm/h ~ 25mm/h, 如图 2-6 所示。

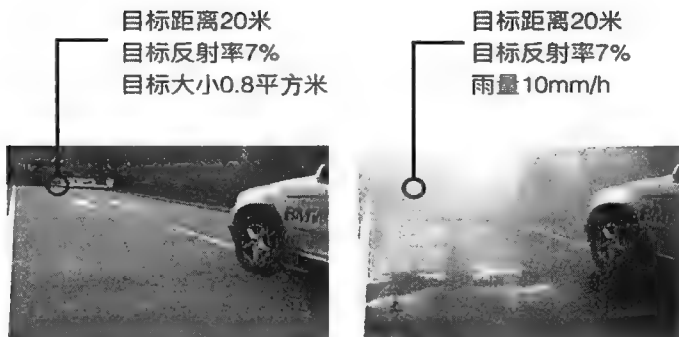


图 2-6 外部环境对 LiDAR 测量的影响

如图 2-7 所示，这里使用了 A 和 B 两个来自不同制造厂的 LiDAR，可以看到随着实验雨量的增大，两种 LiDAR 的最远探测距离都线性下降。雨中或雾中的传播特性最近几年随着激光技术的广泛应用越来越受学术研究界的重视。研究表明，雨和雾都是由小水滴构成的，雨滴的半径直接和其在空中的分布密度直接决定了激光在传播的过程中与之相撞的概率。相撞概率越高，激光的传播速度受到的影响越大。

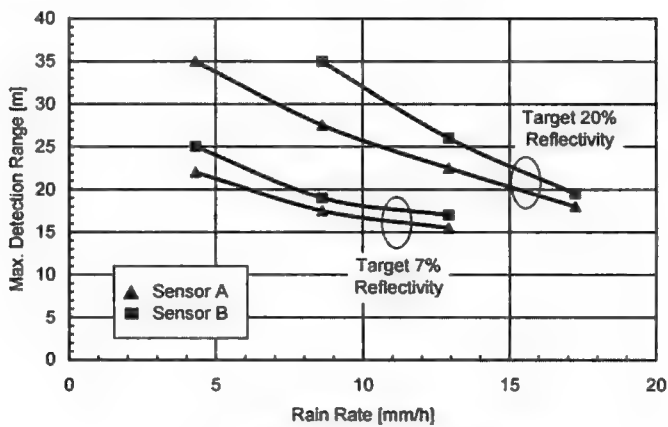


图 2-7 雨量对 LiDAR 测量影响的量化

2.4.2 计算性能的挑战：计算量大

如表 2-1 所示，即使是 16 线的 LiDAR 每秒钟要处理的点也达到了 30 万个。如此大量的数据处理是无人车定位算法和障碍物检查算法的实时性需要面临的一大挑战。例如，之前所说的 LiDAR 给出的原始数据只是反射物体的距离信息，需要对所有的产生的点进

行几何变换,将其转化为位置坐标,这其中至少涉及了4次浮点运算和3次三角函数运算,而且点云在后期的处理中还有大量坐标系转换等更多复杂的运算,这些都对计算资源(CPU、GPU 和 FPGA) 提出了很大的需求。

表 2-1 不同 LiDAR 的每秒计算量比较

型 号	Channel 数量	每秒产生点数
Velodyne HDL-64E	64 Channels	2,200,000
Velodyne HDL-32E	32 Channels	700,000
Velodyne VLP-16	16 Channels	300,000

2.4.3 成本的挑战：造价昂贵

LiDAR 的造价也是要考虑的重要因素之一。上面提到的 Velodyne VLP-16 LiDAR 官网税前售价为 7999 美元,而 Velodyne HDL-64E LiDAR 预售价也在 10 万美元以上。这样的成本要加在本来就没有过高利润的汽车价格中,无疑会大大阻碍无人车的商业化。

2.5 展望未来

尽管无人驾驶技术渐趋成熟,但 LiDAR 始终是一个绕不过去的技术。纯视觉与 GPS/IMU 的定位及避障方案价格虽然低,却还不成熟,很难应用到室外场景中。同时,LiDAR 的价格高居不下,消费者很难承受动辄几十万美元定价的无人车。因此,当务之急就是快速把系统成本特别是 LiDAR 的成本大幅降低。其中一个较有希望的方法是使用较低价的 LiDAR,虽然会损失一些精确度,但可以使用其他低价传感器与 LiDAR 做信息混合,较精准地推算出车辆的位置。换言之,就是通过更好的算法弥补硬件传感器的不足,我们相信这是无人车近期的发展方向,而高精度 LiDAR 的价格由于市场需求大增将会在未来的两三年内出现降幅,为无人车的进一步普及铺路。

2.6 参考资料

[1] B. Schwarz, “LiDAR: Mapping the world in 3D”, *Nature Photon*, vol. 4, pp. 429-430, 2010.

[2] J.Levinson, M.Montemerlo, and S.Thrun, “Map-based precision vehicle localization

in urban environments”, *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

- [3] J.Levinson and S.Thrun, “Robust vehicle localization in urban environments using probabilistic maps”, *ICRA'10*, 2010, pp.4372-4378.
- [4] M.E.El Najjar and P.Bonnifait, “A road-matching method for precise vehicle localization using belief theory and Kalman filtering”, *Autonomous Robots*, vol.19, no.2, pp.173-191, 2005.
- [5] Z. Chong, B. Qin, T. Bandyopadhyay, M. Ang, E. Frazzoli, and D. Rus, “Synthetic 2d LIDAR for precise vehicle localization in 3d urban environment”, *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 2013.
- [6] A. Segal, D. Haehnel, and S. Thrun, “Generalized-ICP”, *Proc. Robot.: Sci. & Syst. Conf.*, Seattle, WA, June 2009.
- [7] I.Baldwin and P.Newman, “Road vehicle localization with 2d pushbroom lidar and 3d priors”, *Proc.IEEE International Conference on Robotics and Automation (ICRA2012)*, Minnesota, USA, May 2012.
- [8] A.Harrison and P.Newman, “High quality 3d laser ranging under general vehicle motion”, *Proc.IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, California, April 2008.

3

GPS 及惯性传感器在无人驾驶中的应用

本章着重介绍全球定位系统及惯性传感器在无人驾驶中的应用。GPS 是当前行车定位不可或缺的技术，但是由于 GPS 的误差、多路径，以及更新频率低等问题，我们不可以只依赖于 GPS 进行定位。相反，惯性传感器拥有很高的更新频率，可以跟 GPS 形成很好的互补。使用传感器融合技术，可以融合 GPS 与惯性传感器数据，各取所长，以达到较好的定位效果。

3.1 无人驾驶定位技术

行车定位是无人驾驶最核心的技术之一，GPS 是当前行车定位不可或缺的技术，在无人驾驶定位中也担负起相当重要的职责。然而无人车是在复杂的动态环境中行驶，尤其在大城市中，GPS 多路径反射的问题会更加明显，这样得到的 GPS 定位信息很容易就有几米的误差。对于在有限宽度上高速行驶的汽车，这样的误差很有可能导致交通事故。因此，必须借助其他传感器辅助定位，增强定位的精度。另外，由于 GPS 的更新频率低（10 Hz），在车辆快速行驶时很难给出精准的实时定位。惯性传感器（IMU）主要是检测和测量加速度与旋转运动的高频（1kHz）传感器，对惯性传感器数据进行处理后我们可以实时得出

车辆的位移与转动信息,但是惯性传感器自身也有偏差与噪声等问题影响结果。通过使用基于卡尔曼滤波的传感器融合技术,我们可以融合 GPS 与惯性传感器数据,各取所长,以达到较好的定位效果。注意,由于无人驾驶对可靠性和安全性要求非常高,所以基于 GPS 及惯性传感器的定位并非无人驾驶里唯一的定位方式,还可以使用 LiDAR 点云与高精地图匹配,以及视觉里程计算法等定位方法,让各种定位法互相纠正以达到更精准的定位。

3.2 GPS 简介

GPS 是当前行车定位不可或缺的技术,在无人驾驶定位中也担负起相当重要的职责。^[1] GPS 系统包括太空中的 32 颗 GPS 卫星;地面上 1 个主控站、3 个数据注入站和 5 个监测站及作为用户端的 GPS 接收机。最少只需其中 3 颗卫星,就能迅速确定用户端在地球上所处的位置及海拔高度。现在,民用 GPS 也可以达到十米左右的定位精度。当前的 GPS 系统使用低频信号,纵使天气不佳仍能让信号保持相当的穿透性。本节介绍 GPS 的运作原理及这个技术的缺陷。

3.2.1 三边测量法定位

如图 3-1 所示, GPS 定位系统是利用卫星基本三角定位原理, GPS 接收装置以量测无线电信号的传输时间来量测距离。由每颗卫星的所在位置,测量每颗卫星至接收器间距离,便可以算出接收器所在位置之三维空间坐标值。使用者只要利用接收装置接收到 3 个卫星信号,就可以定出使用者所在之位置。在实际应用中, GPS 接收装置都是利用 4 个以上卫星信号来定出使用者所在之位置及高度。三角定位的工作原理如下。

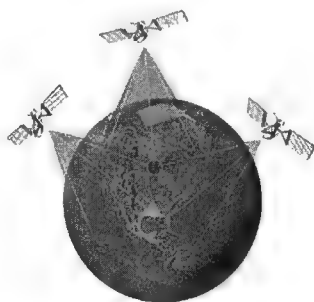


图 3-1 GPS 三边测量法定位原理

(1) 假设测量得出第一颗卫星距离我们 18000 千米,那么可以把当前位置范围限定在距离第一颗卫星 18000 千米半径的地球表面上的任意位置。

(2) 假设测量到第二颗卫星的距离为 20000 千米,那么可以进一步把当前位置范围限定在距离第一颗卫星 18000 千米及距离第二颗卫星 20000 千米的交叉的区域。

(3) 再对第三颗卫星进行测量,通过 3 颗卫星的距离交汇点定位出当前的位置。通常, GPS 接收器会使用第四颗卫星的位置对前 3 颗卫星的位置测量进行确认已达到更好的效果。

3.2.2 距离测量与精准时间戳

理论上,距离测量是个简单的过程,我们只需要用光速乘以信号传播时间便可以得到距离信息。但问题是测量的传播时间但凡有一点误差,都会造成距离上巨大的误差。我们日常使用的时钟是存在一定误差的,如果使用石英钟对传播时间进行测量,那么基于 GPS 的定位会有很大误差。为了解决这个问题,每颗卫星上都安装了原子钟以达到纳米级的精度。为了使卫星定位系统使用同步时钟,需要在所有卫星及接收机上都安装原子钟。原子钟的价格在几万美元,要让每一个 GPS 接收器安装原子钟是不现实的。为了解决这一难题,每一颗卫星上仍然使用昂贵的原子钟,但接收机使用的是经常需要调校的普通石英钟:接收机接收来自四颗或更多卫星的信号并计算自身的误差,因此接收机就可以将自身的时钟调整到统一时间值。

3.2.3 差分 GPS

如上所述,卫星距离测量存在着卫星钟误差与播延迟导致的误差等问题。利用差分技术,我们可以消除或者降低这些误差,让 GPS 达到更高的精度。差分 GPS 的运作原理十分简单,如图 3-2 所示,首先,如果两个 GPS 接收机的距离非常接近,那么两者接收的 GPS 信号将通过几乎同一块大气区域,所以二者的信号将具有非常近似的误差。如果能精确地计算出第一个接收机的误差,那么我们可以利用该计算误差对第二个接收机的结果进行修正。

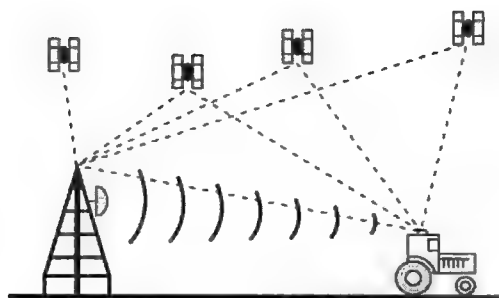


图 3-2 差分 GPS 定位原理

如何精确地计算出第一个接收机的误差呢？可以在已知精准的地点安置参考接收机基准站，安装在基准站上的 GPS 接收机观测 4 颗卫星后便可进行三维定位，计算出基准站的测量坐标。然后，将测量坐标与已知坐标进行对比可以计算出误差。基准站再把误差值发送给方圆 100 千米内的差分 GPS 接收器去纠正它们的测量数据。

3.2.4 多路径问题

如图 3-3 所示，多路径问题是指由于 GPS 信号的反射与折射造成的信号传播时间的误差，从而导致定位的错误^[2]。特别是在城市环境中，空气中有许多悬浮介质对 GPS 信号反射与折射，另外，信号也会在高楼大厦的外墙发生反射与折射，造成距离测量的混乱。目前，高精度的军用差分 GPS 在静态的时候确实可以在“理想”的环境下达到厘米级的精度。这里的“理想”环境是指大气中没有过多的悬浮介质而且测量时 GPS 有较强的接收信号。然而，无人车是在复杂的动态环境中行驶的，尤其在大城市中，GPS 多路径反射的问题会更明显。这样得到的 GPS 定位信息很容易就有几米的误差。对于在有限宽度上高速行驶的汽车，这样的误差很有可能会导致交通事故。

即使有各种问题，GPS 还是一个相对精准的传感器，而且 GPS 的误差不会随着时间的推进而增加。但是 GPS 有一个更新频率低的问题，大概在 10Hz。由于无人车行驶速度快，我们需要实时的精准定位以确保无人车的安全。因此我们必须借助其他的传感器来辅助定位，增强定位的精度。

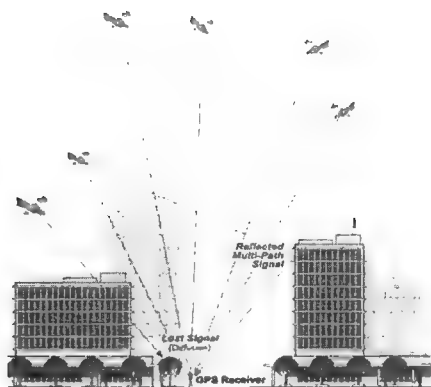


图 3-3 GPS 中存在的多路径问题示意

3.3 惯性传感器简介

惯性传感器 (IMU) 主要是检测和测量加速度与旋转运动的传感器。基础的惯性传感器包括加速度计与角速度计。本节主要讨论基于 MEMS 的六轴惯性传感器，主要由三个轴加速度传感器及三个轴的陀螺仪组成。^[3]

MEMS 惯性传感器分为三个级别：低精度惯性传感器作为消费电子类产品主要用在智能手机，此类传感器售价在 50 美分到几美元不等，测量的误差会比较大。中级惯性传感器主要用于汽车电子稳定系统及 GPS 辅助导航系统，此类传感器售价在几百到上千美元，相对于低端惯性传感器，中级惯性传感器在控制芯片中对测量误差有一定的修正，所以测量结果更准确。但是长时间运行后，累计的误差会越来越大。高精度的惯性传感器作为军用级和宇航级产品，主要要求高精度、全温区、抗冲击等指数，主要用于通信卫星无线、导弹导引头、光学瞄准系统等稳定性应用。此类传感器售价在几十万美元，即便经过长时间运行后，比如跨太平洋洲际导弹，仍然可以达到米级精度。

无人车使用的一般是中低级的惯性传感器，其特点是更新频率高 (1kHz)，可以提供实时位置信息。惯性传感器的致命缺点是它的误差会随着时间的推进而增加，所以我们只能在很短的时间内依赖惯性传感器进行定位。

3.3.1 加速度计

图 3-4 所示为 MEMS 加速度计，它的工作原理就是靠 MEMS 中可移动部分的惯性。

由于中间电容板的质量很大，而且它是一种悬臂构造，当速度变化或者加速度达到足够大时，它所受到的惯性力超过固定或者支撑它的力，这时它会移动，它跟上下电容板之间的距离就会变化，上下电容就会因此变化。电容的变化与加速度成正比。根据不同测量范围，中间电容板悬臂构造的强度或者弹性系数可以设计得不同。另外，如果要测量不同方向的加速度，这个 MEMS 的结构会有很大的不同。电容的变化会被另外一块专用芯片转化成电压信号，有时还会把这个电压信号放大。电压信号在数字化后经过一个数字信号处理过程，在零点和灵敏度校正后输出。

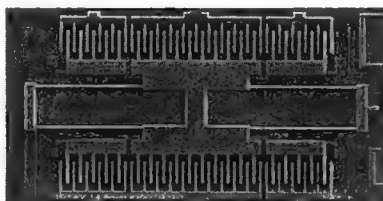


图 3-4 加速度计结构图

3.3.2 角速度计

图 3-5 所示为 MEMS 陀螺仪角速度计 (MEMS gyroscope)，其工作原理主要是利用角动量守恒原理，因此它主要是一个不停转动的物体，它的转轴指向不随承载它的支架的旋转而变化。与加速度计工作原理相似，陀螺仪的上层活动金属与下层金属形成电容。当陀螺仪转动时，它与下面电容板之间的距离就会变化，上下电容也就会因此变化。电容的变化跟角速度成正比，由此我们可以测量当前的角速度。



图 3-5 角速度计示意图

3.3.3 惯性传感器的问题

由于制作工艺的原因，惯性传感器测量的数据通常都会有一定误差。第一种误差是偏

移误差,也就是陀螺仪和加速度计即使在没有旋转或加速的情况下也会有非零的数据输出。要想得到位移数据,我们需要对加速度计的输出生进行两次积分。在两次积分后,即使很小的偏移误差也会被放大,随着时间的推进,这样的偏移误差造成的位移误差会不断积累,最终导致没法再跟踪无人车的位置。第二种误差是比例误差,是指所测量的输出和被检测输入的变化之间的比率。与偏移误差相似,在两次积分后,随着时间的推进,其造成的位移误差会不断积累。第三种误差是背景白噪声,如果不给予纠正,也会导致我们没法再跟踪无人车的位置。

为了纠正这些误差,我们必须对惯性传感器进行校准,找出偏移误差和比例误差,然后使用校准参数对惯性传感器原数据进行修正。但复杂的是惯性传感器的误差也会随着温度变化而变化,所以即使我们校准得再好,随着时间的推进,位移的误差还是会不断积累,所以我们很难单独使用惯性传感器对无人车进行定位。

3.4 GPS 和惯性传感器的融合

如上所述,纵使有多路径等问题,GPS 是个相对精准的定位传感器,但是 GPS 的更新频率低,并不能满足实时计算的要求。而惯性传感器的定位误差会随着运行时间增长,但是由于惯性传感器是高频传感器,在短时间内可以提供稳定的实时位置更新。所以,只要我们找到一个方法能融合这两种传感器的优点,各取所长,就可以得到比较实时与精准的定位。本节介绍如何使用卡尔曼滤波器融合这两种传感器数据。

3.4.1 卡尔曼滤波器简介

卡尔曼滤波器可以从一组有限的、包含噪声的、通过对物体位置的观察序列预测出物体的位置的坐标及速度。卡尔曼滤波器具有很强的鲁棒性,即使对物体位置的观测有误差,根据物体历史状态与当前对位置的观测,我们可以较准确地推算出物体的位置。卡尔曼滤波器运行时主要分两个阶段:预测阶段基于上个时间点的位置信息去预测当前的位置信息;更新阶段通过当前对物体位置的观测去纠正位置预测,从而更新物体的位置。

举个具体例子,假设你家停电没有任何灯光,你想从你家客厅走向卧室。你十分清楚客厅与卧室的相对位置,于是你在黑暗中行走,并试图通过计算步数预测你的当前位置。走到一半时,你摸到了电视机。由于你事先知道电视机在客厅的大致位置,于是可以通过你印象中电视机的位置更正你对当前位置的预测,然后在这个调整过的更准确的位置估计

的基础上继续依靠计算步数向卧室前行。通过摸黑前行，你依靠计算步数与触摸物体最终从客厅走回了卧室，其背后的道理就是卡尔曼滤波器的核心。

3.4.2 多传感器融合

如图 3-6 所示，使用卡尔曼滤波器对惯性传感器与 GPS 数据进行融合与上面给出的例子很相似^{[4][5][6]}。这里的惯性传感器相当于数步数，而 GPS 数据相当于电视等参照物的位置。首先，我们在上一次的位置估算的基础上使用惯性传感器对当前的位置进行实时预测。在得到新的 GPS 数据之前，我们只能通过积分惯性传感器的数据预测当前位置。惯性传感器的定位误差会随着运行时间增长，所以当接收到新的 GPS 数据时，由于 GPS 的数据比较精准，我们可以使用 GPS 数据对当前的位置预测进行更新。通过不断地执行这两个步骤，我们可以取两者所长，对无人车进行准确的实时定位。假设惯性传感器的频率是 1kHz，而 GPS 的频率是 10Hz，那么每两次 GPS 更新之间会使用 100 个惯性传感器数据点进行位置预测。

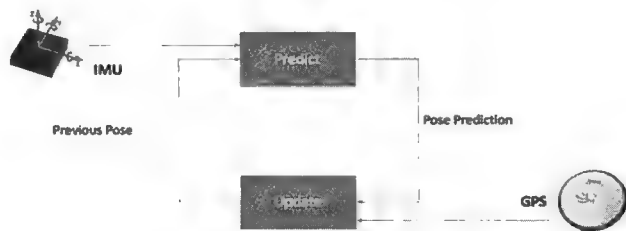


图 3-6 GPS 与 IMU 的传感器融合定位示意

3.5 结论

本章介绍了在无人驾驶场景中如何使用 GPS 与惯性传感器对车辆进行精准定位。这个系统包含了三个部分：第一，相对精准但是低频更新的 GPS；第二，高频更新但是精度随着时间流逝而越发不稳定的惯性传感器；第三，上述两种传感器基于卡尔曼滤波器数学模型的融合。由于无人驾驶对可靠性和安全性的要求非常高，所以除 GPS 与惯性传感器外，通常还会使用 LiDAR 点云与高精地图匹配，以及视觉里程计算法等定位方法，让各种定位法互相纠正以达到更精准的定位。

3.6 参考资料

- [1] Samama, Global Positioning Technologies And Performance, *John Wiley & Sons, Inc.*, Hoboken, New Jersey, 2008.
- [2] T.Kos, I.Markezic, and J.Pokrajcic, “Effects of multipath reception on GPS positioning performance”, *ELMAR, 2010 PROCEEDINGS.IEEE*, 2010, pp.399-402.
- [3] O.J. Woodman, An Introduction to Inertial Navigation; UCAM-CL-TR-696; Computer Laboratory, *University of Cambridge*: Cambridge, UK, 2007.
- [4] H.Carvalho, P.DelMoral, A.Monin, and G.Salut, “Optimal nonlinear filtering in GPS/INS integration”, *IEEE Transactions on Aerospace and Electronic Systems*, vol.33, no.3, pp.835-850, 1997.
- [5] A.Mohamed and K.Schwarz, “Adaptive Kalman filtering for INS/GPS” , *Journal of Geodesy*, vol.73, no.4, pp.193-203, 1999.
- [6] H.H.Qi and J.B.Moore, “Direct Kalman filtering approach for GPS/INS integration”, *IEEE Transactions on Aerospace and Electronic Systems*, vol.38, no.2, pp.687-693, 2002.

4

基于计算机视觉的无人驾驶感知系统

本章着重介绍基于计算机视觉的无人驾驶感知系统。在现有的无人驾驶系统中，LiDAR 是当仁不让的感知主角，但是由于 LiDAR 的成本高等因素，业界有许多讨论是否可以使用成本相对较低的摄像头去承担更多的感知任务。本章会探索基于视觉的无人驾驶感知方案。首先，要验证一个方案是否可行，我们需要一个标准的测试方法：被广泛使用的无人驾驶视觉感知数据集 KITTI。然后，会讨论计算机视觉在无人车场景中使用到的具体技术，包括 Optical Flow 和立体视觉、物体的识别和跟踪，以及视觉里程计算法。

4.1 无人驾驶的感知

在无人驾驶技术中，感知是最基础的部分，没有对车辆周围三维环境的定量感知，就犹如人没有了眼睛，无人驾驶的决策系统就无法正常工作。为了安全与准确地感知，无人车驾驶系统使用了多种传感器，其中可以被广义地划分为“视觉”的有超声波雷达、毫米波雷达、LiDAR，以及摄像头。超声波雷达由于反应速度和分辨率的特性主要用于倒车雷达。激光雷达和毫米波雷达则主要承担了中长距测距和环境感知的功能。

其中，激光雷达在测量精度和速度上表现得更出色，是厘米级的高精度定位中不可或

缺的部分，但是其制造成本极其昂贵，并且其精度易受空气中悬浮物的干扰。相较而言，毫米波雷达则更能适应较恶劣的天气，抗悬浮物干扰性强，但是仍需防止其他通信设备和雷达之间的电磁波干扰。可见光的摄像头视觉数据分析与处理基于发展已久的传统计算机视觉领域，其通过摄像头采集到的二维图像信息推断三维世界的物理信息，现通常应用于交通信号灯识别和其他物体识别。那么最常见的、成本相对低廉的摄像头解决方案能否在无人驾驶应用中承担更多的感知任务呢？

4.2 KITTI 数据集

KITTI 数据集是由 KIT 和 TTIC 在 2012 年开始的一个合作项目，网站为 <http://www.cvlibs.net/datasets/kitti/>，这个项目的目的是建立一个具有挑战性的，来自真实世界的测试集^{[1][2][3][4]}。如图 4-1 所示，他们使用的数据采集车配备了如下设备。

- 一对 140 万像素的彩色摄像头，Point Grey Flea 2 (FL2-14S3C-C)，10Hz 采集频率。
- 一对 140 万像素的黑白摄像头，Point Grey Flea 2 (FL2-14S3M-C)，10Hz 采集频率。
- 一个激光雷达，Velodyne HDL-64E。
- 一个 GPS/IMU 定位系统，OXTS RT 3003。

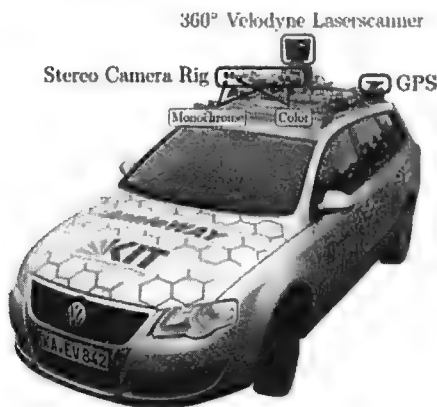


图 4-1 装载各种传感器的数据采集车

这辆车在卡尔斯鲁厄的高速和城区的多种交通环境下收集了数据，用激光雷达提供的的数据作为 ground truth，建立了面向多个测试任务的数据集。

(1) Stereo/Optical Flow 数据集：数据集由图片对组成。一个 Stereo 图片对是两个摄

摄像头在不同的位置同时拍摄的,Optical Flow 图片对是同一个摄像头在相邻时间点拍摄的。训练数据集有 194 对,测试数据集有 195 对,大约 50%的像素有确定的偏移量数据,如图 4-2 所示。



图 4-2 stereo/optical flow 数据集示意图

(2) 视觉里程测量数据集:数据集由 22 个 Stereo 图片对序列组成,一个 4 万多帧,覆盖 39.2 公里的里程,如图 4-3 所示。



图 4-3 视觉里程测量数据集示意图

(3) 三维物体检测数据集:手工标注,包含轿车、厢车、卡车、行人、自行车者、电车等类别,用三维框标注物体的大小和朝向,有多种遮挡情况,并且一张图片通常有多个物体实例,如图 4-4 所示。

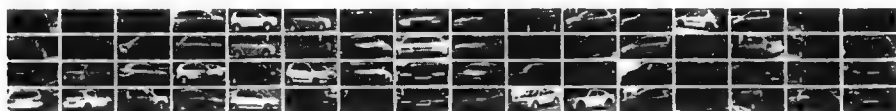


图 4-4 三维物体检测数据集示意图

(4) 物体追踪数据集:手工标注,包含 21 个训练序列和 29 个测试序列,主要追踪的目标类型是行人和轿车,如图 4-5 所示。



图 4-5 物体追踪数据集示意图

(5) 路面和车道检测数据集:手工标注,包含未标明车道、标明双向单车道和标明双向多车道三种情况,289 张训练图片和 290 张测试图片,ground truth 包括路面(所有车道)和自车道,如图 4-6 所示。



图 4-6 路面和车道检测数据集示意图

KITTI 数据集与以往计算机视觉领域的数据集相比有以下特点。

- 由无人车上常见的多种传感器收集，用 LiDAR 提供高精度的三维空间数据，有较好的 ground truth。
- 更接近实际情况，而不是用计算机图形学技术生成的。
- 覆盖了计算机视觉在无人车驾驶上应用的多个方面。

由于这些特点，越来越多的研究工作基于这个数据集，一个新的算法在这个数据集上的测试结果有较高的可信度。

4.3 计算机视觉能帮助无人车解决的问题

计算机视觉在无人车上的使用有一些比较直观的例子，比如交通标志和信号灯的识别（谷歌）；高速公路车道的检测定位（特斯拉）。现在，基于 LiDAR 信息实现的一些功能模块其实也可以用摄像头基于计算机视觉来实现。下面介绍计算机视觉在无人车驾驶上的几个应用前景。当然，这只是计算机视觉在无人车上的部分应用，随着技术的发展，越来越多的基于摄像头的算法会让无人车的感知更准确、更快速、更全面。

计算机视觉在无人车场景中解决的最主要问题可以分为两大类：物体的识别与跟踪，以及车辆本身的定位。**物体的识别与跟踪**：通过深度学习的方法，无人车可以识别在行驶途中遇到的物体，比如行人、空旷的行驶空间、地上的标志、红绿灯，以及旁边的车辆等。由于行人及旁边的车辆等物体都是在运动的，我们需要跟踪这些物体以达到防止碰撞的目的，这就会涉及 Optical Flow 等运动预测的算法。**车辆本身的定位**：通过基于拓扑与地标的算法，或者是基于几何的视觉里程计算法，无人车可以实时确定本身的位置，以满足自主导航的需求。

4.4 Optical Flow 和立体视觉

物体的识别与跟踪,以及车辆本身的定位都离不开底层的 **Optical Flow** 与立体视觉技术。在计算机视觉领域, **Optical Flow** 是图片序列或者视频中像素级的密集对应关系,例如在每个像素上估算一个二维的偏移矢量,得到的 **Optical Flow** 以二维的矢量场表示^[5]。立体视觉则是从两个或更多的视角得到的图像中建立对应关系。这两个问题有高度的相关性,一个是基于单个摄像头在连续时刻的图像,另一个是基于多个摄像头在同一时刻的图片。解决这类问题时有两个基本假设。

- 不同图像中的对应点都来自物理世界中同一点的成像,所以“外观”相似。
- 不同图像中的对应点集合的空间变换基本满足刚体条件,或者说空间上分割为多个刚体的运动。从这个假设我们自然得到 **Optical Flow** 的二维矢量场是片状平滑的结论。

2016 年 6 月,美国拉斯维加斯召开的 CVRP 大会上, Urtasun 教授和她的学生改进了深度学习中的 Siamese 网络,用一个内积层代替了拼接层,把处理一对图片的时间从一分钟左右降低到不到一秒。^[6]

如图 4-7 所示,这个 Siamese 结构的神经网络分左右两部分,分别是一个多层的卷积神经网络和两个卷积神经网络共享网络权重。**Optical Flow** 的偏移矢量估计问题转化为一个分类问题,输入是两个 9×9 的图片块,输出是 128 或者 256 个可能的偏移矢量 y 。通过从已知偏移矢量的图片对中抽取的图片块输入到左右两个卷积神经网络,然后最小化 cross-entropy (如下面公式所示)。

$$\min_w \left[- \sum_{i, y_i} p_{gt}(y_i) \log p_i(y_i, w) \right]$$

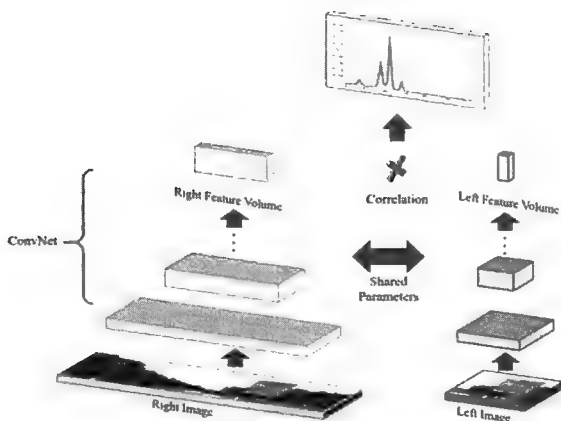


图 4-7 Siamese 结构的深度神经网络分层示意

我们可以用监督学习的方法训练整个神经网络。

- i 是像素的指标。
- y_i 是像素 i 可能的偏移矢量。
- p_{gt} 是一个平滑过的目标分布, 用来给一两个像素的预估误差反馈一个非 0 的概率, gt 表示 ground truth。
- $p_i(y_i, w)$ 是神经网络输出的给定 w 时 y_i 的概率。

在 KITTI 的 stereo 2012 数据集上, 这样的算法可以在 0.34 秒的时间里完成计算, 并达到相当出色的精度, 偏移估计误差在 3~4 像素左右, 对大于 3 像素的偏移估计误差在 8.61 像素, 表现优于其他低速度的算法。

在得到每个像素上 y_i 的分布后, 我们还需要加入空间上的平滑约束, 本节试验了三种方法。

- 最简单直接的 5×5 窗口平均。
- 加入了相邻像素 y 一致性的半全局块匹配 (semi global block matching)。
- 超像素+3 维斜面。

这些平滑方法一起, 能把偏移估计的误差再降低大约 50%, 这样就得到了一个比较准确的 2 位偏移矢量场。基于它, 我们能够得到图 4-8 所示场景的三维深度/距离估计。这样的信息对无人车自动驾驶非常重要。

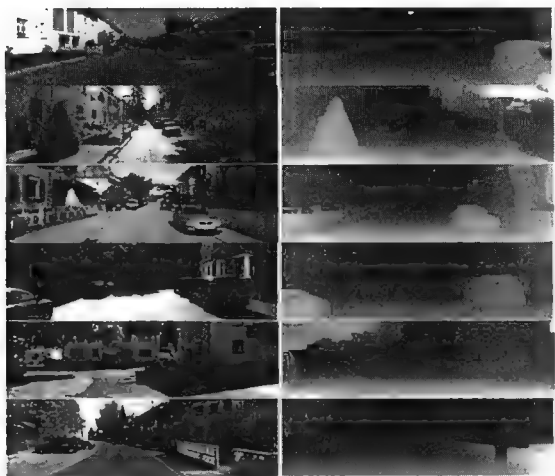


图 4-8 深度信息图示意

4.5 物体的识别与追踪

从像素层面的颜色、偏移和距离信息到物体层面的空间位置和运动轨迹，是无人车视觉感知系统的重要功能。无人车的感知系统需要实时的识别和追踪多个运动目标 (multi-object tracking, MOT)，例如车辆和行人。^[7] 物体识别问题是计算机视觉的核心问题之一，最近几年由于深度学习的革命性发展，计算机视觉领域大量使用卷积神经网络，物体识别的准确率和速度得到了很大提升，但总的来说物体识别算法的输出一般是有噪声的：物体的识别有可能不稳定，物体可能被遮挡，可能有短暂误识别等。MOT 问题中流行的 tracking-by-detection 方法就要解决这样一个难点：如何基于有噪声的识别结果获得鲁棒的物体运动轨迹。在 ICCV 2015 会议上，斯坦福大学的研究者发表了基于马尔可夫决策过程 (MDP) 的 MOT 算法来解决这个问题，下面我们就详细介绍这个工作。

运动目标的追踪用一个 MDP 来建模，如图 4-9 所示。

- 运动目标的状态： $s \in S = S_{\text{active}} \cup S_{\text{tracked}} \cup S_{\text{lost}} \cup S_{\text{inactive}}$ ，这几个子空间各自包含无穷多个目标状态。被识别到的目标首先进入 active 状态，如果是误识别，目标进入 inactive 状态，否则进入 tracked 状态。处于 tracked 状态的目标可能进入 lost 状态，处于 lost 状态的目标可能返回 tracked 状态，或者保持 lost 状态，或者在足够长时间之后进入 inactive 状态。

- 作用 $a \in A$ ，所有作用都是确定性的。
- 状态变化函数 $T: S \times A \rightarrow S$ 定义了状态 s 和作用 a 下目标状态变为 s' 。
- 奖励函数 $R: S \times A \rightarrow R$ 定义了作用 a 之后到达状态 s 的即时奖励，这个函数是从训练数据中学习的。
- 规则 $\pi: S \rightarrow A$ 决定了在状态 s 采用的作用 a 。

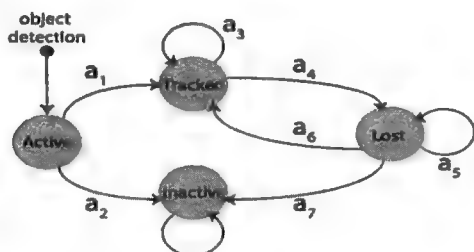


图 4-9 MDP 状态转化示意图

如图 4-10 所示，这个 MDP 的状态空间变化如下。

- 在 active 状态下，物体识别算法提出的物体候选通过一个线下训练的支持向量机（SVM），判断下一步的作用是 a_1 还是 a_2 ，这个 SVM 的输入是候选物体的特征向量、空间位置大小等，它决定了在 S_{active} 中的 MDP 规则 π_{active} 。
- 在 tracked 状态下，一个基于 tracking-learning-detection 追踪算法的物体线上外观模型被用来决定目标物体是否保持在 tracker 状态还是进入 lost 状态。这个外观模型（Appearance Model）使用当前帧中目标物体所在的矩形（Bounding Box）作为模板（Template），所有在 tracked 状态下收集的物体外观模板在 lost 状态下被用来判断目标物体是否回到 tracked 状态。另外在 tracked 状态下，物体的追踪使用上述外观模型模板，矩形范围内的 Optical Flow 和物体识别算法提供的候选物体和目标物体的重合比例决定是否保持在 tracked 状态，如果是，那么目标物体的外观模板自动更新。
- 在 lost 状态下，如果一个物体保持 lost 状态超过一个阈值帧数，就进入 inactive 状态；物体是否返回 tracked 状态由一个基于目标物体和候选物体相似性特征向量的分类器决定，对应了 S_{lost} 中的 π_{lost} 。

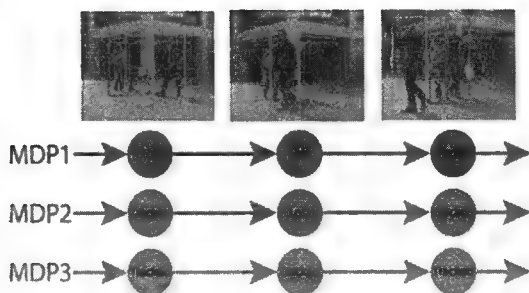


图 4-10 MDP 状态转换实例

这个基于 MDP 的算法在 KITTI 数据集的物体追踪评估中达到了业界领先水平。

4.6 视觉里程计算法

基于视觉的定位算法有两大分类：一种是基于拓扑与地标的算法，另一种是基于几何的视觉里程计算法。基于拓扑与地标的算法把所有的地标组成一个拓扑图，然后当无人车监测到某个地标时，便可以大致推断出自己所在的位置。基于拓扑与地标的算法相对于基于几何的方法容易些，但是要求预先建立精准的拓扑图，比如将每个路口的标志物做成地标。基于几何的视觉里程计算法计算比较复杂，但是并不需要预先建立精准的拓扑图，这种算法可以在定位的同时扩展地图。本节我们将着重介绍视觉里程计算法。

视觉里程计算法主要分为单目及双目两种。纯单目视觉里程算法存在的主要问题是无法推算出观察到的物体大小，所以使用者必须假设或者推算出一个初步的大小，或者通过与其他传感器结合（比如陀螺仪）进行准确的定位。双目的视觉里程计算法通过左右图 triangulation 计算出特征点的深度，然后从深度信息中推算出物体的大小。图 4-11 所示为双目视觉里程计算法的具体计算流程。

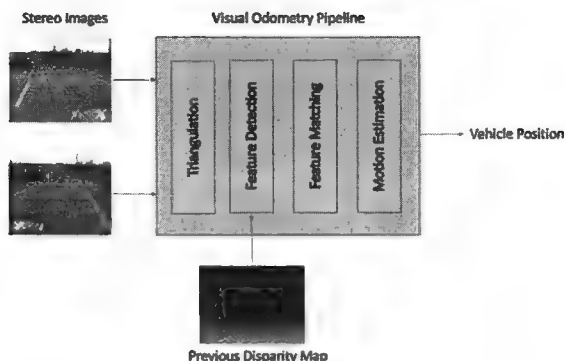


图 4-11 视觉里程计算法原理图

(1) 双目摄像机抓取左右两图。

(2) 双目图像经过 triangulation 产生当前帧的 disparity map。

(3) 提取当前帧与之前帧的特征点，如果之前帧的特征点已经提取好了，那么我们可以直接使用之前帧的特征点。特征点提取可以使用 Harris Corner Detector。

(4) 对比当前帧与之前帧的特征点，找出帧与帧之间的特征点对应关系。具体可以使用 RANSAC 算法。

(5) 根据帧与帧之间的特征点对应关系，推算出两帧之间车辆的运动。这个推算是最小化两帧之间的 reprojection error 实现的。

(6) 根据推算出的两帧之间车辆的运动，以及之前的车辆位置，计算出最新的车辆位置。

通过以上视觉里程计算法，无人车可以实时推算出自己的位置，进行自主导航，但是纯视觉定位计算的一个很大的问题是算法本身对光线相当敏感。在不同的光线条件下，同样的场景不能被识别。特别在光线较弱时，图像会有很多噪点，极大地影响了特征点的质量。在反光的路面，这种算法也很容易失效。这也是影响视觉里程计算法在无人驾驶场景普及的一个主要原因。可能的解决方法是在光线条件不好的情况下，更加依赖根据车轮及雷达返回的信息进行定位，我们会在后面章节中详细讨论这部分内容。

4.7 结论

本章，我们探索了基于视觉的无人驾驶感知方案。首先，要验证一个方案是否可行，

我们需要一个标准的测试方法。为此我们介绍了无人驾驶的标准 KITTI 数据集。在有了标准的数据集后,研究人员可以开发基于视觉的无人驾驶感知算法,并使用数据集对算法进行验证。然后,我们详细了解了计算机视觉的 Optical Flow 和立体视觉,物体的识别和跟踪,以及视觉里程计算法等技术,以及这些技术在无人驾驶场景的应用。视觉主导的无人车系统是目前研究的前沿,虽然目前各项基于视觉的技术还没完全成熟,但我们相信在未来五年,如果 LiDAR 的成本不能降下来,基于摄像机的视觉感知会逐步取代 LiDAR 的功能,为无人车的普及打好基础。

无人驾驶可能是计算机视觉发展的一次难得的机遇,无人车产业爆发带来的资源,无人车收集的大量真实世界的数据和 LiDAR 提供的高精度三维信息可能意味着计算机视觉将要迎来“大数据”和“大计算”带来的红利,数据的极大丰富和算法的迭代提高相辅相成,会推动计算机视觉研究的前进,从而在无人驾驶中起到更加不可或缺的作用。

4.8 参考资料

- [1] Andreas Geiger, Philip Lenz and Christoph Stiller and Raquel Urtasun, “Vision meets Robotics: The KITTI Dataset”, *International Journal of Robotics Research (IJRR)*, 2013.
- [2] Andreas Geiger, Philip Lenz and Raquel Urtasun, “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite”, *CVPR 2012*.
- [3] Moritz Menze and Andreas Geiger, “Object Scene Flow for Autonomous Vehicles”, *CVPR 2015*.
- [4] Jannik Fritsch, Tobias Kuehnl and Andreas Geiger, “A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms”, *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.
- [5] Florian Raudies, “Optic flow”, *Scholarpedia*, 2013, 8(7):30724.
- [6] W. Luo, A. Schwing and R. Urtasun, “Efficient Deep Learning for Stereo Matching”, *CVPR 2016*.
- [7] Yu Xiang, Alexandre Alahi and Silvio Savarese. “Learning to Track: Online Multi-Object Tracking by Decision Making”, *ICCV 2015*.

5

卷积神经网络在无人驾驶中的应用

本章着重介绍卷积神经网络在无人驾驶中的应用。无人驾驶的感知部分作为计算机视觉的领域范围，也不可避免地成为卷积神经网络发挥作用的舞台。本章将深入介绍卷积神经网络在无人驾驶 3D 感知与物体检测中的应用。

5.1 CNN 简介

卷积神经网络 (Convolutional Neural Network, CNN) ^{[1][2][3][4][5]} 是一种适合使用在连续值输入信号上的深度神经网络，比如声音、图像和视频。它的历史可以回溯到 1968 年 Hubel 和 Wiesel 在动物视觉皮层细胞中发现的对输入图案的方向选择性和平移不变性，这个工作为他们赢得了诺贝尔奖。时间推进到 20 世纪 80 年代，随着神经网络研究的深入，研究人员发现对图片输入做卷积操作和生物视觉中的神经元接受局部 (Receptive Field) 内的输入有相似性，那么在神经网络中加上卷积操作也就成了自然而然的事情。当前的 CNN 相比通常的深度神经网络 (Deep Neural Network, DNN)，它的特点主要如下。

(1) 一个高层的神经元只接受某些低层神经元的输入，这些低层神经元处于二维空间中的一个邻域，通常是一个矩形。这个特点受了生物神经网络中 Receptive Field 概念的

启发。

(2) 同一层中不同神经元的输入权重共享, 这个特点可以认为是利用了视觉输入中的平移不变性, 不光大幅度减少了 CNN 模型的参数数量, 还加快了训练速度。

由于 CNN 在神经网络的结构上针对视觉输入本身特点做的特定的设计, 所以它是计算机视觉领域使用 DNN 时的不二选择。在 2012 年 CNN 一举打破了 ImageNet 这个图像识别竞赛的世界纪录之后, 计算机视觉领域发生了天翻地覆的变化, 各种视觉任务都放弃了传统方法, 启用 CNN 构建新的模型。无人驾驶的感知部分作为计算机视觉的领域范围的一部分, 也不可避免地成为 CNN 发挥作用的舞台。

5.2 无人驾驶双目 3D 感知

在无人车感知中, 对周围环境的 3D 建模是重中之重。激光雷达能提供高精度的 3D 点云, 但密集的 3D 信息就需要摄像头的帮助了。人用两只眼睛获得立体的视觉感受, 同样的道理能让双目摄像头提供 3D 信息。假设两个摄像头间距为 B , 空间中一点 P 到两个摄像头所成图像上的偏移 (disparity) 为 d , 摄像头的焦距为 f , 那么我们可以计算 P 点到摄像头的距离为:

$$z = \frac{B}{d} f$$

所以为了感知 3D 环境得到 z , 我们需要通过双目摄像头的两张图像 I_l 和 I_r 得到 d , 通常的做法是基于局部的图片匹配:

$$I_l(p) \Rightarrow I_r(p + d)$$

由于单个像素的值可能不稳定, 所以需要利用周围的像素和平滑性假设 $d(x, y) \approx d(x + \alpha, y + \beta)$ (假设 α 和 β 都较小), 这样求解 d 就变成了一个最小化问题:

$$\min_d D(q, d) = \min_d \sum_{q \in N(p)} \| I_l(q) - I_r(q + d) \|$$

这和 Optical Flow 任务想要解决的是非常类似的问题, 只不过 (I_l, I_r) 变成了 (I_t, I_{t+1}) , 所以下面将要介绍的算法, 对于两者都适用。

5.2.1 MC-CNN

现在来看看 Matching-Cost CNN 算法，^[6] 这个算法使用了一个 CNN 来计算上式的右侧 matching cost，MC-CNN 的网络结构如图 5-1 所示。

这个网络的输入是两个图片的一小块，输出是这两块不匹配的概率，相当于一个 cost 函数，当两者匹配时为 0，不匹配时最大可能为 1。通过对一个给定的图片位置搜索可能的 d 取值，找到最小的 CNN 输出，就得到了这一点局部的偏移估算。MC-CNN 算法接下来做了如下后期处理。

(1) Cross-based cost aggregation: 基本思想是对邻近的像素值相似的点的偏移求平均，提高估计的稳定性和精度。

(2) Semi-global matching: 基本思想是邻近的点的平移应该相似，加入平滑约束并求偏移的最优值。

(3) 插值和图片边界修正：提高精度，填补空白。

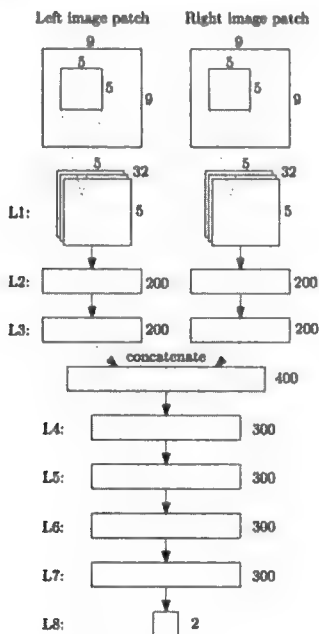


图 5-1 MC-CNN 的网络结构

最终算法效果如图 5-2 所示。

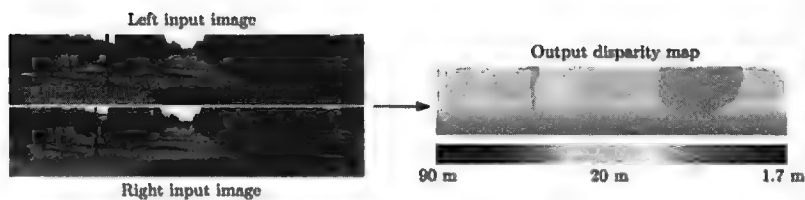


图 5-2 MC-CNN 的算法效果

MC-CNN 虽然使用了 CNN，但仅限于计算匹配程度，后期的平滑约束和优化都是必不可少的，那么有没有可能使用 CNN 一步到位呢？FlowNet 就是这样做的。

5.2.2 FlowNet

为了实现端到端的模型结构，我们需要用 CNN 实现特征提取，匹配打分和全局优化等功能。FlowNet 采取了 encoder-decoder 的框架，把一个 CNN 分成了收缩和扩张两个部分，^[7] 如图 5-3 所示。

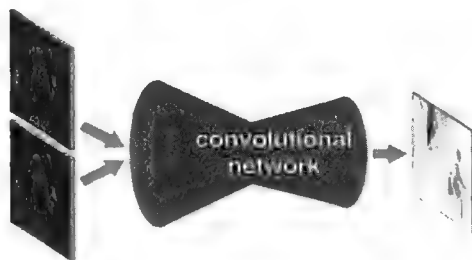


图 5-3 encoder-decoder 算法框架

在收缩部分 FlowNet 提出了两种可能的模型结构。

（1）FlowNetSimple：把两幅图片叠起来输入到一个“线性”的 CNN 中，输出是每个像素的偏移量。这个模型的弱点是计算量大，而且无法考虑全局的优化手段，因为每个像素的输出是独立的。

（2）FlowNetCorr：先对两幅图片分别进行特征的提取，然后通过一个相关层把两个分支合并起来并继续下面的卷积层运算。这个相关层的计算和卷积层类似，只是没有了学习到的特征权重，而是由两个分支得到的隐层输出相乘求和，如图 5-4 所示。

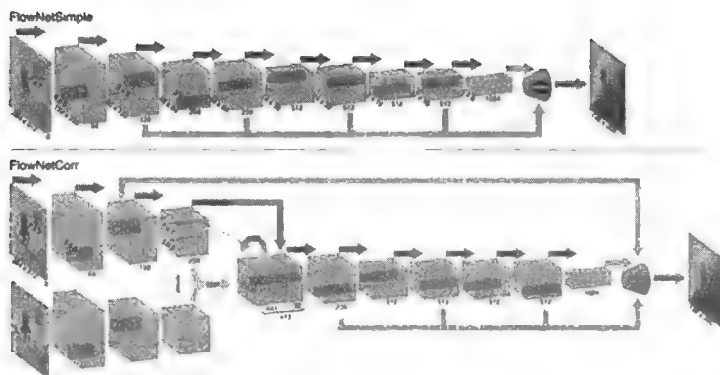


图 5-4 FlowNetSimple 与 FlowNetCorr 原理图

FlowNet 网络收缩部分不仅减少了 CNN 的计算量，同时起到了在图像平面上聚合信息的作用，但这也导致了分辨率的下降。于是 FlowNet 在网络扩张部分使用了“up convolution”来提高分辨率，注意这里不仅使用了上一层的低分辨率输出，还使用了网络收缩部分的相同尺度的隐层输出，如图 5-5 所示。

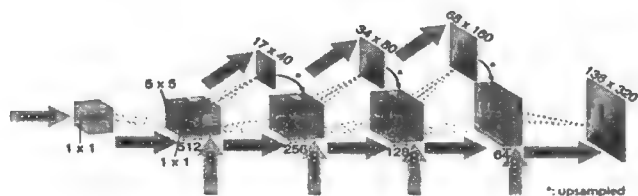


图 5-5 FlowNet 网络扩张原理图

FlowNet 算法在常见的公开数据集上都获得了不错的效果，尤其值得一提的是它的速度很快。

5.3 无人驾驶物体检测

物体检测技术是无人驾驶感知必不可少的部分。自从 2012 年 CNN 在图片分类问题上取得突破，物体检测这个问题自然成为了 CNN 应用的下一个目标，使用 CNN 的物体检测算法层出不穷，我们只挑选有代表性的几个算法做介绍。

在 CNN 在物体识别领域里大行其道之前，通常的做法是类似于 DPM (Deformable

Parts Model) 这样的解决方案: 在图像上抽取局部特征的组合作为模板, 比如基于图像的空间梯度的 HOG 特征, 为了能够处理形变、遮挡等变化, 我们建立一个“弹性”的结构把这些“刚性”的部分组合起来, 最后加上一个分类器判断物体是否出现。这样的算法一般复杂度较高, 需要大量的经验, 而且改进和优化难度较大。CNN 的到来改变了一切。

R-CNN 系列算法^{[8][9][10]}是一个两段式的算法, 它把物体识别这个问题分为两方面。

- 物体可能所在区域的选择: 输入一张图片, 由于物体在其中的位置大小有太多可能性, 我们需要一个高效的方法找出它们, 这里的重点是在区域个数的一定上限下, 尽可能地找到所有的物体, 关键指标是召回率。
- 候选区域的识别: 给定了图片中的一块矩形区域, 识别其中的物体并修正区域大小和长宽比, 输出物体类别和更“紧”的矩形框。这里的重点在识别的精度。

在了解了算法的大致架构后, 我们来看看算法的具体实现, 这里我们主要描述 R-CNN 这一系列算法的最新版: Faster R-CNN, 它对应上面的两步分为 RPN (Region Proposal Network) 和 Fast R-CNN, 我们将分别介绍。

5.3.1 RPN

我们称物体可能所在区域为候选, RPN^[8]的功能就是最高效地产生这样一个候选列表。如图 5-6 所示, RPN 选择使用 CNN 为基础, 图片通过多个 (比如 4 个) 卷积层进行特征提取, 在最后一个卷积层输出的特征图上使用一个 3×3 的滚动窗口连接到一个 256 或者 512 维的全连接隐层, 最后再分支到两个全连接层, 一个输出物体类别, 一个输出物体的位置大小。为了能够使用不同的物体大小和长宽比, 在每一个位置上我们考虑三个尺度 (128×128 、 256×256 、 512×512) 和三个长宽比 (1:1、1:2、2:1) 一共 9 种组合。这样一个 1000×600 的图片上我们考虑了 $(1000/16) \times (600/16) \times 9 \sim 20,000$ 种位置大小和长宽比的组合, 由于我们使用 CNN 计算, 这一步耗时不多。最后, 我们根据空间重叠程度去掉冗余的候选区域, 一张图片获得 2000 个左右的物体可能区域。

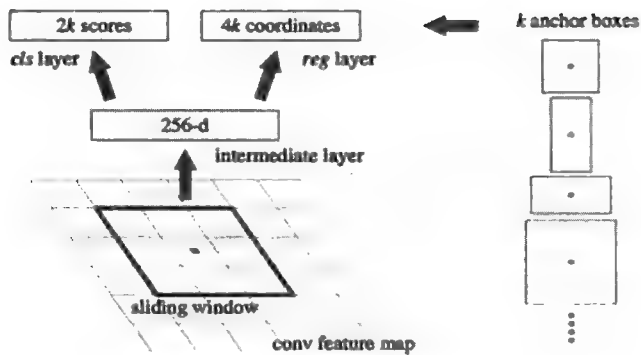


图 5-6 RPN 算法原理图

5.3.2 Fast R-CNN

在候选区域分类阶段，我们使用的是基于全连接的神经网络，如图 5-7 所示的右侧部分。

根据图 5-7 左侧的特征提取部分我们可以重用 RPN 中的 CNN 计算结果，这大大节约了计算时间，能达到 5~17 帧每秒的速度。

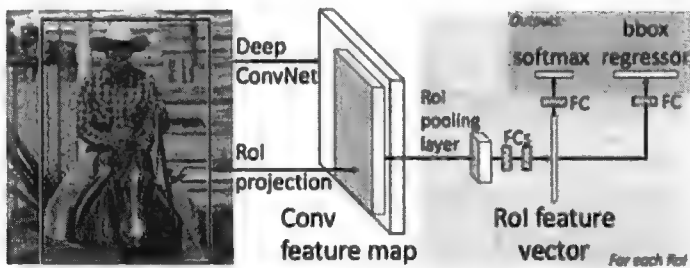


图 5-7 Fast R-CNN 算法原理图

5.3.3 MS-CNN

虽然 Faster R-CNN 算法大名鼎鼎，但在物体尺度变化很大的场景（比如无人驾驶）中还有提升的空间，Multi-scale CNN（MS-CNN）^[11]正是针对这个问题的一个尝试。CNN 的层级结构由于 pooling 层的存在自然形成了和不同尺度的对应关系。那我们为什么不把对物体的检测放到 CNN 的不同层里去呢？这正是 MS-CNN 的想法。

在选择物体候选区域阶段，MS-CNN 使用了如图 5-8 所示的网络结构，我们看到如果

把 CNN 网络里的卷积层看成一个大树的“主干”，那么在 conv3、conv4 和 conv5 三个卷积层之后这个网络都长出了“分支”，每个“分支”都连接了一个检测层，负责一定的尺度范围，这样多个“分支”一起，就能覆盖比较宽的物体尺度范围，达到我们的目的。

在候选区域识别阶段，如图 5-9 所示，我们让上一阶段多个检测层的输出特征图分别输入到一个子网络里，这里有几个值得注意的细节。

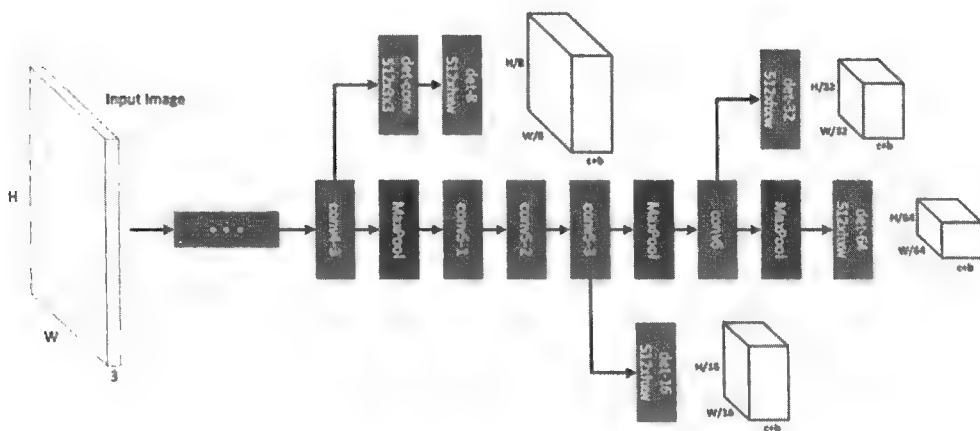


图 5-8 MS-CNN 分层模型示意图

（1）第一层是个“deconvolution”层，目的是提高特征图的分辨率，保证物体检测的准确率，特别是对尺度偏小的物体来说。

（2）Deconvolution 之后，在抽取物体特征的时候（外框），我们还同时抽取了物体周边的信息（内框），这些“上下文”信息对识别准确率的提高有明显帮助。

总的来说，MS-CNN 和 Faster R-CNN 相比，优势是识别的准确度有很大提高，尤其在物体尺度变化的情况下，比如 KITTI 数据集里的行人和自行车，但 Faster R-CNN 还是有速度的优势。

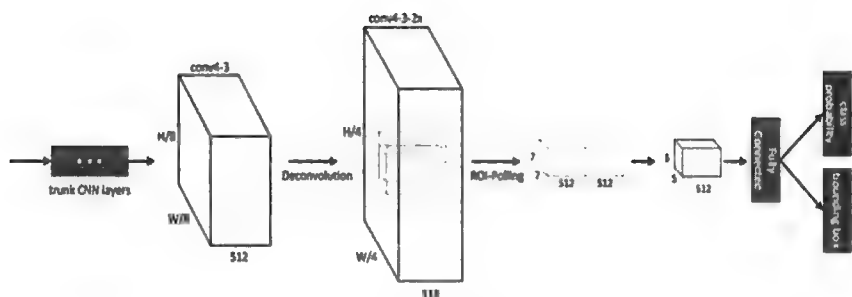


图 5-9 MS-CNN 算法原理图

5.3.4 SSD

虽然 Faster R-CNN 的速度相比之前的 R-CNN 已经有了很大提高,但还是达不到实时的要求。Single Shot Detector (SSD)^[12]就是一个能够实时运行,有更佳准确度的算法,最近人气很高。SSD 沿用了滑动窗口的思想,通过离散化物体的位置、大小和长宽比,使用 CNN 高效计算了各种可能的物体情况,从而达到高速检测物体的目的,如图 5-10 所示。

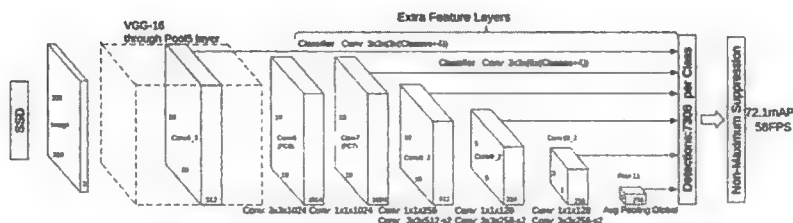


图 5-10 SSD 算法原理图

SSD 使用了 VGG-16 网络做底层的图片特征提取,通过取消生成候选区域、图片缩放和特征图采样的步骤,一步到位判断物体位置和分类,SSD 是一种高速的物体检测算法。

在 VGG 网络的基础上,SSD 加入了逐步变小的卷积层,这些不同尺度的卷积层分别使用 3×3 大小的卷积核进行物体位置偏移和分类的判断,使得 SSD 能够检测到不同大小的物体。

5.4 结论

无人驾驶的感知部分的主要功能是计算机视觉领域研究的相关问题,因此将不可避免地成为 CNN 发挥作用的舞台。CNN 在无人驾驶中的应用主要包括 3D 感知与物体检测。在 3D 感知中使用到的网络包括 MC-CNN 与 FlowNet, 在物体检测中使用到的网络包括 Faster R-CNN、MSCNN 与 SSD。本章详细介绍了各种网络的优缺点,希望对读者选择网络时有帮助。

5.5 参考资料

- [1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition", *Proceedings of the IEEE*, vol. 86, No.11, pp. 2278-2324, Nov. 1998.
- [2] Shan Sung LIEW. Gender classification: A convolutional neural network approach.
- [3] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction teraction, and functional architecture in the cat's visual cortex".
- [4] Jake Bouvrie. Notes on Convolutional Neural Networks.
- [5] Y. LeCun, L. Bottou, G. Orr, and K. Muller, "Efficient BackProp", in *Neural Networks: Tricks of the trade*, (G. Orr and Muller K., eds.), 1998.
- [6] Jure Zbontar, Yann LeCun, Computing the Stereo Matching cost with a Convolutional Neural Network, *Computer Vision and Pattern Recognition (CVPR)*, 2015 IEEE Conference, 7-12 June 2015.
- [7] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, Thomas Brox, FlowNet: Learning Optical Flow with Convolutional Networks, *IEEE International Conference on Computer Vision (ICCV)*, *Computer Vision and Pattern Recognition (CVPR)*, 2015 IEEE Conference, 7-12 June 2015.
- [8] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster R-CNN: Towards

- Real-Time Object Detection with Region Proposal Networks, arXiv:1506.01497.
- [9] Ross Girshick, Fast R-CNN, arXiv:1504.08083.
- [10] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, *CVPR*, 2014.
- [11] Zhaowei Cai, Quanfu Fan , Rogerio S. Feris , and Nuno Vasconcelos, A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection.
- [12] W Liu, D Anguelov, D Erhan, C Szegedy, S Reed, C Fu, A C. Berg, SSD: Single Shot MultiBox Detector, *ECCV 2016*.

6

增强学习在无人驾驶中的应用

本章着重介绍增强学习在无人驾驶中的应用。增强学习的目的是通过和环境交互学习到如何在相应的观测中采取最优行为。相比传统的机器学习,增强学习有以下优势:首先,由于不需要标注的过程,增强学习可以更有效地解决环境中存在的特殊情况。其次,增强学习可以把整个系统作为一个整体的系统,从而对其中的一些模块更加鲁棒。最后,增强学习可以比较容易地学习到一系列的行为。这些特性十分适用于无人驾驶中的决策过程,本章将深入探讨增强学习如何在无人驾驶决策过程中发挥作用。

6.1 增强学习简介

增强学习 (Reinforcement Learning) 是最近几年机器学习领域的最新进展。增强学习的目的是通过和环境交互学习到如何在相应的观测中采取最优行为。^[1] 行为的好坏可以通过环境给的奖励确定。不同的环境有不同的观测和奖励。例如,驾驶中环境的观测是摄像头和激光雷达采集到的周围环境的图像和点云,以及其他传感器的输出 (如行驶速度、GPS 定位和行驶方向)。驾驶中的环境的奖励根据任务的不同,可以通过到达终点的速度、舒适度和安全性等指标确定。

增强学习和传统机器学习的最大区别是增强学习是一个闭环学习的系统,增强学习算法选取的行为会直接影响环境,进而影响该算法之后从环境中得到的观测。传统的机器学习把收集训练数据和模型学习作为两个独立的过程。以学习一个人脸分类的模型为例,传统机器学习的方法首先需要我们雇佣标注者标注一批人脸图像的数据,然后我们在这些数据中学习模型,最后我们可以把训练出来的人脸识别模型在现实的应用中进行测试。如果发现测试的结果不理想,那么我们需要分析模型中存在的问题,并且试着从数据收集或者模型训练中寻找问题的原因,然后从这些步骤中解决这些问题。对于同样的问题,增强学习采用的方法是通过在人脸识别的系统中尝试进行预测,并且通过用户反馈的满意程度调整自己的预测,从而统一收集训练数据和模型学习的过程。增强学习和环境交互过程的框图如图 6-1 所示。

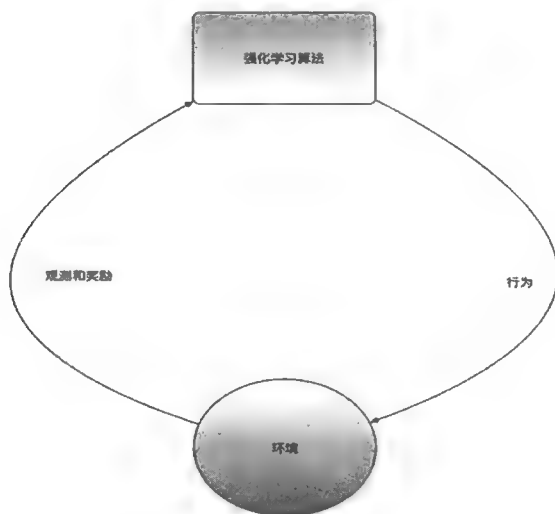


图 6-1 增强学习和环境交互示意图

增强学习存在着很多传统机器学习不具备的挑战。首先,因为在增强学习中没有确定的在每一个时刻应该采取哪个行为的信息,所以增强学习算法必须通过探索各种可能的行为才能判断出最优的行为。如何在可能行为数量较多的情况下有效地进行探索是增强学习中最重要的问题之一。其次,在增强学习中一个行为不仅可能影响到当前时刻的奖励,而且还可能影响到之后所有时刻的奖励。在最坏的情况下,一个好行为不会在当前时刻获得奖励,而会在很多步都执行正确后才能得到奖励。在这种情况下,增强学习判断出奖励和很多步之前的行为有关是非常难的。

无人驾驶中的感知模块不可能做到完全可靠。前段时间, Tesla 的无人驾驶事故就是在强光的环境中感知模块失效导致的。增强学习可以做到,即使在某些模块失效的情况下也能做出稳妥的行为。增强学习可以比较容易地学习到一系列的行为。无人驾驶中需要执行一系列正确的行为才能成功的驾驶。如果只有标注数据,学习到的模型每个时刻偏移了一点,到最后可能会偏移非常多,产生毁灭性的后果。增强学习能够学会自动修正偏移。

综上所述,增强学习在无人驾驶中有广泛的前景。本章会介绍增强学习的常用算法及其在无人驾驶中的应用。希望能够激发读者对这个领域的探索性工作。

6.2 增强学习算法

增强学习中的每个时刻 $t \in \{0, 1, 2, \dots\}$ 中,我们的算法和环境通过执行行为 a_t 进行交互,可以得到观测 s_t 和奖励 r_t 。一般情况下,我们假设环境是存在马尔科夫性质的,也就是环境的变化完全可以通过状态转移概率 $Pass' = Pr\{s_{t+1}=s' | s_t=s, a_t=a\}$ 刻画出来。也就是说,环境下一时刻的观测值与给定当前时刻的观测值和行为,以及与之之前所有时刻的观测值和行为都没有关系。而环境早 $t+1$ 时刻返回的奖励在当前状态和行为确定下的期望可以表示为: $Ras = E\{r_{t+1} | s_t=s, a_t=a\}$ 。增强学习算法在每一个时刻执行行为的策略可以通过概率 $\pi(s, a, \theta) = Pr\{a_t=a | s_t=s; \theta\}$ 表示,其中 θ 是需要学习的策略的参数。我们需要学习到最优的增强学习策略,也就是学习到能够取得最高奖励的策略^{[2][3]}。

$$\rho(\pi) = E\{\sum_{t=1}^{\infty} \gamma^{t-1} r_t | s_0, \pi\}$$

其中 γ 是增强学习中的折扣系数,用来表示在之后时刻得到的奖励的折扣。同样的奖励,获得的时刻越早,增强学习系统所感受到的奖励越高。

同时,我们可以按照如下方式定义 Q 函数。Q 函数 $Q_{\pi}(s, a)$ 表示的是在状态为 s , 执行行为 a 之后的时刻都使用策略 π 选择行为能够得到的奖励。我们能够学习到准确的 Q 函数,那么使 Q 函数最高的行为就是最优的行为。

$$Q_{\pi}(s, a) = E\{\sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k} | s_t=s, a_t=a, \pi\} = E_{s'}[r + \gamma Q_{\pi}(s', a') | s, a, \pi]$$

增强学习的目的就是在给定的任意环境下,通过对环境进行探索,学习到最佳的策略函数 π 最大化 $\rho(\pi)$ 。下面的章节中我们会简单介绍常用的增强学习的算法,包括 REINFORCE 算法和 Deep Q-Learning 算法。

6.2.1 REINFORCE 算法

REINFORCE 算法是最简单的增强学习算法。REINFORCE 算法的基本思想通过在一个回合里执行当前的策略直到一个回合结束（比如游戏结束），根据得到的奖励可以计算出当前策略的梯度。我们可以用这个梯度更新当前的策略得到新的策略。在下面的回合，我们再用新的策略重复这个过程，一直到计算出的梯度足够小为止。最后得到的策略就是最优策略。

假设我们当前的策略的概率是 $\pi(\theta|x)=Pr\{a_t=a|s_t=s;\theta\}$ （ θ 是策略的参数）。每个回合，算法实际执行的行为 a_t 是按照概率 $\pi(x)$ 采样得到的。算法在当前回合的时刻 t 获得的奖励用 r_t 表示。那么，策略的梯度可以通过以下公式计算。

$$\nabla_{\theta} p(\pi) = \sum_{t=1}^T \nabla_{\theta} \log[\pi(a_t|s_t; \theta)] R_t$$

其中 $\pi(a_t|s_t; \theta)$ 是策略在观测到 s_t 时选择 a_t 的概率。 $R_t = \sum_{i=t}^T \gamma^i (r_i - tr_t)$ 是算法在采取了当前策略之后获得的总的折扣后的奖励。为了减少预测出的梯度的方差。我们一般会使用 $(R_t - b_t)$ 代替 R_t 。 b_t 一般等于 $E\pi[R_t]$ ，也就是当前 t 时刻的环境下使用策略 π 之后能获得的折扣后奖励的期望。

计算出方差之后，我们可以使用 $\theta = \theta + \nabla_{\theta} p(\pi)$ 更新参数得到新的策略。

REINFORCE 算法的核心思想是通过从环境中获得的奖励判断执行的行为的好坏。如果一个行为执行之后获得的奖励比较高，那么算出的梯度也会比较高，这样在更新后的策略中该行为被采样到的概率也会比较高。反之，对于执行之后获得奖励比较低的行为，因为计算出的梯度低，更新后的策略中该行为被采样到的概率也会比较低。通过在这个环境中反复地执行各种行为，REINFORCE 可以大致准确地估计出各个行为的正确梯度，从而对策略中各个行为的采样概率做出相应的调整。

作为最简单的采样算法，REINFORCE 算法得到了广泛的应用，例如学习视觉的注意力机制和学习序列模型的预测策略都用到了 REINFORCE 算法。事实证明，在模型相对简单、环境的随机性不强的环境下，REINFORCE 算法可以达到很好的效果。

但是，REINFORCE 算法也存在它的问题。首先，在 REINFORCE 算法中，执行了一个行为之后的所有奖励都被认为是因为这个行为产生的。这显然是不合理的。虽然在执行了策略足够多的次数然后对计算出的梯度进行平均之后，REINFORCE 有很大的概率计算出正确的梯度，但在实际中，出于效率的考虑，同一个策略在更新之前不可能在环境中执

行太多次。在这种情况下, REINFORCE 计算出的梯度有可能会比较大的误差。其次, REINFORCE 算法有可能会收敛到一个局部最优点中。如果我们已经学到了一个策略, 这个策略中大部分的行为都是以近似 1 的概率采样到的, 那么即使这个策略不是最优的, REINFORCE 算法也很难学习到如何改进这个策略, 因为我们完全没有执行其他采样概率为 0 的行为, 无法知道这些行为的好坏。最后, REINFORCE 算法只有在环境存在回合概念时才能够使用。如果环境不存在回合的概念, REINFORCE 算法将无法使用。

最近, DeepMind 提出了使用 Deep Q-learning 算法的学习策略, 克服了 REINFORCE 算法的缺点, 在 Atari 游戏学习这样的复杂的任务中取得了令人惊喜的效果。

6.2.2 Deep Q-Learning

Deep Q-Learning 是一种基于 Q 函数的增强学习算法。该算法对于复杂的、每步行为之间存在较强相关性的环境有很好的学习效果。Deep Q-Learning 的学习算法的基础是 Bellman 公式^[4]。我们在 6.2.1 节已经介绍了 Q 函数的定义。

如果我们学习到了最优的行为对应的 Q 函数 $Q(s,a)$, 那么这个函数应该满足下面的 Bellman 公式。

$$Q(s,a)=E_{s'}[r+\gamma\max_a'Q(s,a)|s,a]$$

另外, 如果我们学习到了最优的行为对应的 Q 函数 $Q(s,a)$, 那么我们在每一时刻得到了观察 s_t 之后, 可以选择使得 $Q(s,a)$ 最高的行为作为执行的行为 a_t 。

我们可以用一个神经网络计算 Q 函数, 用 $Q(s,a;w)$ 来表示, 其中 w 是神经网络的参数。我们希望我们学习出来的 Q 函数满足 Bellman 公式, 因此可以定义下面的损失函数。这个函数的 Bellman 公式的 L2 误差如下。

$$L(w)=E\{[r+\gamma\max_a'Q(s',a';w)-Q(s,a;w)]^2\}$$

其中 r 是在 s 的观测执行行为 a 后得到的奖励, s' 是执行行为 a 之后下一个时刻的观测。这个公式的前半部分 $r+\gamma\max_a'Q(s',a';w)$ 也被称为目标函数。我们希望预测出的 Q 函数能够 and 通过这个时刻得到的奖励和下一个时刻状态得到的目标函数尽可能接近。通过这个损失函数, 我们可以计算出如下梯度:

$$\partial L(w)/\partial w=E\{[r+\gamma\max_a'Q(s',a';w)-Q(s,a;w)]\partial Q(s,a;w)/\partial w\}$$

我们可以通过计算出的梯度, 使用梯度下降算法更新我们的参数 w 。

使用深度神经网络逼近 Q 函数存在很多问题。首先，在一个回合内采集到的各个时刻的数据是存在着相关性的。因此，如果我们使用了一个回合内的全部数据，那么我们计算出的梯度是有偏的。其次，由于取出使 Q 函数最大的行为这个操作是离散的，即使 Q 函数变化很小，我们得到的行为也可能差别很大。这个问题会导致训练时我们的策略出现震荡。最后， Q 函数的动态范围有可能很大，并且我们很难预先知道 Q 函数的动态范围。因为当我们对一个环境没有足够了解时，很难计算出这个环境中可能得到的最大奖励。这个问题可能会使 Q -Learning 的工程中的梯度很大，导致训练不稳定。

首先，Deep Q -Learning 算法是使用了经验回放的算法。这个算法的基本思想是记住算法在这个环境中执行的历史信息。这个过程和人类的学习过程类似。人类在学习执行行为的策略时，不会只通过当前执行的策略的结果进行学习，还会利用之前的历史执行的策略的经验进行学习。因此，经验回放算法将算法在一个环境中所有的历史经验都存放起来。在学习的时候，可以从经验中采样出一定数量的跳转信息 $(s_t, a_t, r_{t+1}, s_{t+1})$ ，也就是当下所处的环境信息，然后利用这些信息计算出梯度学习模型。因为不同的跳转信息是从不同的回合中采样出来的，所以它们之间不存在强相关性。这个采样过程还可以解决同一个回合中的各个时刻的数据的相关性问题。

而且，Deep Q -Learning 算法使用了目标 Q 网络解决学习过程中的震荡问题。我们可以定义一个目标 Q 网络 $Q(s', a'; w-)$ ，这个网络的结构和用来执行的 Q 网络的结构完全相同，唯一的不同就是使用的参数 $w-$ 不同。我们的目标函数可以通过目标 Q 网络计算。

$$r + \gamma \max_a Q(s', a'; w-)$$

目标 Q 网络的参数在很长时间内保持不变，每当在 Q 网络学习了一定的时间之后，可以用 Q 网络的参数 w 替换目标 Q 网络的参数 $w-$ ，这样目标函数会在很长的时间里保持稳定，可以解决学习过程中的震荡问题。

最后，为了防止 Q 函数的值太大导致梯度不稳定，Deep Q -Learning 算法对奖励设置了最大值和最小值（一般设置为 $[-1, +1]$ ）。我们会把所有的奖励缩放到这个范围，这样算法计算出的梯度会更稳定。

Q -Learning 算法的框图如图 6-2 所示。

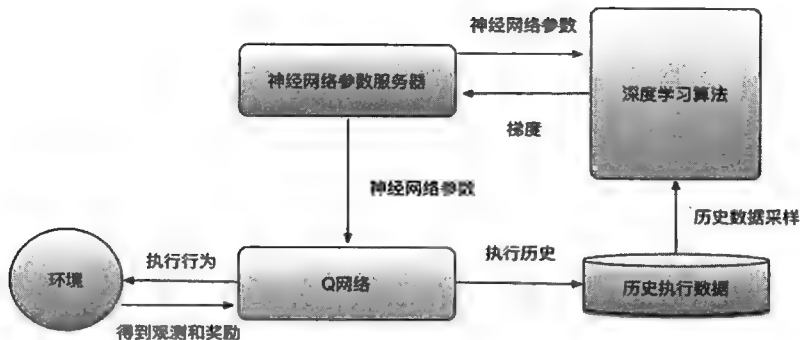


图 6-2 Q-Learning 算法原理图

因为使用了深度神经网络来学习 Q 函数，Deep Q-Learning 算法可以直接以图像作为输入学习复杂的策略，其中一个例子是学习 Atari 游戏。Atari 游戏是计算机游戏的早期形式，一般图像比较粗糙，但是要玩好需要对图像进行理解，并且执行出复杂的策略，例如躲避、发射子弹、走迷宫等。一些 Atari 游戏的例子如图 6-3 所示，我们注意到其中包含了一个简单的赛车游戏。^[5]

Deep Q-Learning 算法在没有任何额外知识的情况下，完全以图像和获得的奖励进行输入，在大部分 Atari 游戏中都大大超过了人类的性能。这在没有深度学习或者增强学习时完全是不可能完成的任务。Atari 游戏是第一个用 Deep Q-Learning 解决了其他算法都无法解决的问题，充分显示了将深度学习和增强学习结合的优越性和前景。



图 6-3 Atari 游戏示例

6.3 使用增强学习帮助决策

现有的深度增强学习解决的问题中我们执行的行为一般只对环境有短期影响。例如，在 Atari 的赛车游戏中，我们只需要控制赛车的方向和速度让赛车沿着跑道行驶，并且躲避其他赛车就可以获得最优的策略，但是对于更复杂决策的情景，我们无法只通过短期的奖励得到最优的策略，一个典型的例子是走迷宫。在走迷宫这个任务中，判断一个行为是否是最优的行为无法从短期的奖励得到，只有当最终走到终点时才能得到奖励。在这种情况下，直接学习出正确的 Q 函数是非常困难的。我们只有结合基于搜索的算法和基于增强学习的算法才能有效地解决这类问题。

基于搜索的算法一般是通过搜索树实现的。搜索树既可以解决一个玩家在环境中探索的问题（例如走迷宫），也可以解决多个玩家竞争的问题（例如围棋）。我们以围棋为例，讲解搜索树的基本概念。围棋游戏有两个玩家，分别由白子和黑子代表。一个围棋棋盘中心的交叉点是可以下子的地方。两个玩家分别在棋盘上下白子和黑子，一旦一片白子或黑子被相反的颜色子包围，那么这片子就会被提掉，重新成为空白的区域。游戏的最后，当所有的空白区域都被占领或者包围时，占领和包围的区域比较大的一方获胜。

在围棋这个游戏中，我们从环境中得到的观测 s_t 是棋盘的状态，也就是白子和黑子的分布。我们执行的行为是所下的白子或者黑子的位置。我们最后得到的奖励可以根据游戏是否取胜得到。取胜的一方得到的奖励是+1，失败的一方得到的奖励是-1。这个游戏的进程可以通过如图 6-4 所示的搜索树表示。搜索树中的每个节点对应着一种棋盘的状态。每一条边对应着一个可能的行为。黑棋先行，树的根节点对应的是棋盘的初始状态 s_0 。 a_1 和 a_2 对应黑棋的两种可能的落子位置（在实际的围棋中，可能的行为远比两种多），每个行为 a_i 对应着一个新的棋盘的状态 si_1 。接下来该白棋走，白棋同样有两种走法 b_1 和 b_2 ，对于每个棋盘的状态 si_1 ，两种不同的走法又会生成两种不同的状态。如此往复，一直到游戏结束，我们就可以在游戏的叶子节点中获得游戏结束时黑棋获得的奖励。我们可以通过这些奖励获得最佳的状态。

$$\nabla w = \alpha [G_t - v(s_t; w)] \nabla_w v(s_t; w)$$

在这个式子中， G_t 表示的是在 t 时刻之后获得的奖励。因为在围棋这个游戏中，我们只在最后时刻获得奖励，所以 G_t 对应的是最后获得的奖励。我们也可以使用类似 Q-Learning 的方式用 TD 误差更新参数。

$$\nabla w = \alpha [v(s_{t+1}; w) - v(s_t; w)] \nabla_w v(s_t; w)$$

在围棋这个游戏中，我们只在最后时刻获得奖励，一般使用 REINFORCE 算法的更新方式的效果比较好。在学习出一个好的近似最小最大值函数之后，我们可以大大地加快搜索的效率，这和人学习围棋的过程类似。人在学习围棋的过程中会对特定的棋行形成感觉，能够一眼就判断出棋行的好坏，而不用对棋的发展进行推理。这就是通过学习近似最小最大值函数加速搜索的过程。

通过学习近似最小最大值函数，Google DeepMind 在围棋领域取得了突飞猛进的进展。在 2016 年 3 月进行的比赛中，DeepMind 的 AlphaGo 以 4 比 1 的比分战胜了围棋世界冠军李世石。AlphaGo 的核心算法就是利用历史棋局和自己对弈，从而学习近似最小最大值函数的算法^[6]。AlphaGo 的成功充分展示了增强学习和搜索的结合使用在解决涉及长期规划问题的潜力。需要注意的是，现有的将增强学习和搜索结合的算法只能用于确定性的环境中。确定性的环境中给定一个观测和一个行为，下一个观测是确定的，并且这个转移函数是已知的。在环境非确定，并且转移函数未知的情况下，如何将增强学习和搜索结合是增强学习领域中没有解决的问题。

6.4 无人驾驶的决策介绍

无人驾驶的人工智能包含了感知、决策和控制三个方面。感知指的是如何通过摄像头和其他传感器的输入解析出周围环境的信息，例如有哪些障碍物、障碍物的速度和距离、道路的宽度和曲率等。这个部分是无人驾驶的基础，是当前无人驾驶研究的重要方向。控制是指当我们有了一个目标，例如右转 30 度，如何通过调整汽车的机械参数达到这个目标。这个部分已经有相对比较成熟的算法能够解决，不在本文的讨论范围之内。本节，我们着重讲解无人驾驶的决策部分。

无人驾驶的决策是指给定感知模块解析出的环境信息如何控制汽车的行为达到驾驶的目标。例如，汽车加速、减速、左转、右转、换道、超车都是决策模块的输出。决策模

块不仅需要考虑到汽车的安全性和舒适性,保证尽快到达目标地点,还需要在旁边的车辆恶意的情况下保证乘客的安全。因此,决策模块一方面需要对行车的计划进行长期规划,另一方面需要对周围车辆和行人的行为进行预测。而且,无人驾驶中的决策模块对安全性和可靠性有严格的要求。现有的无人驾驶的决策模块一般是根据规则构建的。虽然基于规则的构建可以应付大部分的驾驶情况,对于驾驶中可能出现的各种各样的突发情况,基于规则的决策系统不可能枚举到所有突发情况。我们需要一种自适应的系统来应对驾驶环境中出现的各种突发情况。

基于规则的决策系统大部分可以用有限状态机表示。例如,无人驾驶的高层行为可以分为向左换道、向右换道、跟随和紧急停车。决策系统根据目标可以决定执行高层行为。根据需要执行的高层行为,决策系统可以用相应的规则生成底层行为。基于规则的决策系统的主要缺点是缺乏灵活性。对于所有的突发情况,基于规则的决策系统都需要写一个决策。这种方式很难对所有的突发系统面面俱到。

6.4.1 无人驾驶模拟器

无人驾驶的决策过程中,模拟器起着非常重要的作用。决策模拟器负责对环境中的常见场景进行模拟,例如车道情况、路面情况、障碍物分布和行为、天气等。同时还可以将真实场景中采集到的数据进行回放。决策模拟器的接口和真车的接口保持一致,这样可以保证在真车上使用的决策算法可以直接在模拟器上运行。除了决策模拟器之外,无人驾驶的模拟器还包含了感知模拟器和控制模拟器,用来验证感知和控制模块。^[7]这些模拟器不在本节的讨论氛围之内。

无人驾驶模拟器的第一个重要的功能是验证功能。在迭代决策算法的过程中,我们需要比较容易地衡量算法的性能。例如,我们需要确保新的决策算法能够在常见的场景中正确安全地运行。我们还需要对新的决策算法在常见场景的安全性、快捷性、舒适性进行打分。我们不可能每次更新算法时都在实际的场景中进行测试,这时有一个能可靠反映真实场景的无人驾驶模拟器是非常重要的。

模拟器的另一个重要功能是进行增强学习。通过在模拟器里模拟出各种突发情况,增强学习算法可以利用其在这些突发情况中获得的奖励学习如何应对这些突发情况。这样,我们只要能够模拟出足够的突发情况,我们的增强学习算法就可以学习到对应的突发情况的处理方法,而不用每种突发情况都单独写规则处理。而且,我们的模拟器也可以根据之前增强学习对于突发情况的处理结果,尽量产生出当前的增强学习算法无法解决的突发情

况，从而增强学习的效率。

综上所述，无人驾驶模拟器对决策模块的验证和学习都有着至关重要的作用，是无人驾驶领域的核心技术。如何创建出能够模拟出真实场景、覆盖大部分突发情况，并且和真实的汽车接口兼容的模拟器是无人驾驶研发的难点之一。

6.4.2 增强学习在无人驾驶中的应用和展望

增强学习在无人驾驶中有很高的前景。我们在 TORCS 模拟器中使用增强学习进行了探索性的工作。TORCS 是一个赛车的模拟器。玩家在这个模拟器中的任务是超过其他的 AI 车，以最快的速度达到终点。虽然 TORCS 中的任务和真实的无人驾驶的任务还有很大的区别，但是由于其中算法的评估非常容易进行，TORCS 现在常用于研究无人驾驶中的增强学习算法。TORCS 模拟器的截图如图 6-5 所示。增强学习算法一般可以以前方和后方看到的图像作为输入，也可以以环境的状态作为输入（例如自己的速度、离赛道边缘的距离和跟其他车的距离）。

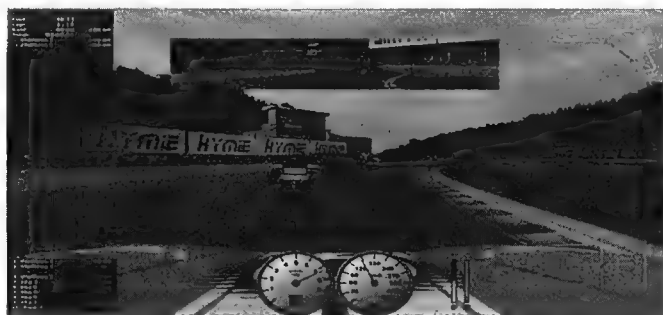


图 6-5 TORCS 模拟器运行状态截图

我们这里使用了环境的状态作为输入，使用 Deep Q-Learning 作为学习算法学习。环境的奖励定义为在单位时刻车辆沿跑道的前进距离。另外，如果车出了跑道或者和其他车辆相撞，会得到额外的惩罚。环境的状态包括了车辆的速度、加速度、离跑道的左右边缘的距离、和跑道的切线的夹角、在各个方向上最近的车的距离等。车的行为包括向上换挡、向下换挡、加速、减速、向左打方向盘、向右打方向盘等。

与普通的 Deep Q-Learning 相比，我们做了以下改进。首先，我们使用了多步的 TD 算法进行更新。多步的 TD 算法能够比单步的算法每次学习时看到更多的执行步数，因此也能够更快地收敛。其次，我们使用了 Actor-Critic 的架构。Actor-Critic 将算法的策略函

数和值函数分别用两个网络表示, 这样的表示有两个优点。

(1) 策略函数可以使用监督学习的方式进行初始化学习。

(2) 在环境比较复杂时, 学习值函数非常困难。把策略函数和值函数分开学习可以降低策略函数学习的难度。

使用了改进后的 Deep Q-Learning 算法, 我们学习到的策略在 TORCS 中可以实现沿跑道行走、换道、超车等行为。基本达到 TORCS 环境中的基本驾驶的需要。Google DeepMind 直接使用图像作为输入, 也获得了很好的效果, 但是训练的过程要慢很多。

现有的增强学习算法在无人驾驶的模拟环境中获得了很有希望的结果, 但是可以看到, 如果需要增强学习真正能够在无人驾驶的场景下应用, 增强学习算法还需要有很多改进。第一个改进方向是增强学习的自适应能力。现有的增强学习算法在环境的性质发生改变时, 需要试错很多次才能学习到正确的行为。而人在环境发生改变的情况下, 只需要很少的试错就可以学习到正确的行为。如何只用非常少量的样本学习到正确的行为是增强学习能够实用的重要条件。

第二个重要的改进方向是模型的可解释性。现在的增强学习中的策略函数和值函数都是由深度神经网络表示的。深度神经网络的可解释性比较差。由于可解释性差, 在实际使用中出了问题很难找到问题的原因, 也比较难排查。在无人驾驶这种人命关天的任务中, 无法找到问题的原因是完全无法接受的。

第三个重要的改进方向是推理和想象的能力。很多时候, 人在学习的过程中不需要有一定的推理和想象的能力。例如, 在驾驶时, 人们不用自己真正尝试, 也知道危险的行为会带来毁灭性的后果, 这是因为人类对这个世界有一个足够好的模型来推理和想象出相应行文可能会发生的后果。这种能力不仅对增强学习算法在存在危险行为环境中的表现非常重要, 在安全的环境中也可以大大加快算法收敛的速度。

只有在这些方向做出了实质性的突破, 增强学习才能真正使用到无人驾驶或者是机器人这种重要的任务场景中。希望更多的有志之士能够投身于增强学习的研究, 为人工智能的发展贡献出自己的力量。

6.5 参考资料

- [1] Richard S. Sutton and Andrew G. Barto, “Reinforcement Learning: An Introduction”, First Edition, *MIT Press*, Cambridge, MA, 1998.
- [2] Ronald J. Williams. A class of gradient-estimating algorithms for reinforcement learning in neural networks. *In Proceedings of the IEEE First International Conference on Neural Networks*, San Diego, CA, 1987.
- [3] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229-256, 1992.
- [4] Bellman, On the Theory of Dynamic Programming, *Proceedings of the National Academy of Sciences*, 1952.
- [5] Mnih, Volodymyr, et al. “Playing atari with deep reinforcement learning”, *arXiv preprint arXiv:1312.5602* (2013).
- [6] Silver, David, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser et al. “Mastering the game of Go with deep neural networks and tree search”. *Nature* 529, no. 7587 (2016): 484-489.
- [7] Wymann, Bernhard, et al. “TORCS, the open racing car simulator”, Software available at <http://torcs.sourceforge.net> (2000).

无人驾驶的规划与控制

7.1 规划与控制简介

无人车作为一个复杂的软硬件结合系统,其安全可靠运行需要车载硬件、传感器集成、感知、预测,以及控制规划等多个模块的协同配合工作。作者认为最关键的部分是感知预测和控制规划的紧密配合。这里的控制规划(Planning & Control)在广义上可以划分成无人车路由寻径(Routing)、行为决策(Behavioral Decision)、动作规划(Motion Planning),以及反馈控制(Feedback Control)等几个部分,如图 7-1 所示。

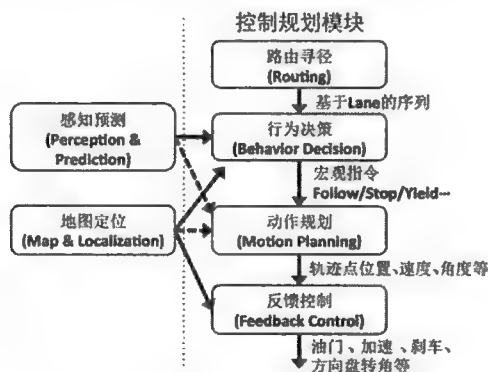


图 7-1 无人车软件系统控制规划示意

控制规划模块的最上游是路由寻径（Routing）模块，其作用在简单意义上可以理解为实现无人车软件系统内部的导航功能，即在宏观层面上指导无人车软件系统的控制规划模块按照什么样的道路行驶，从而实现从起始点到目的地点。值得注意的是，这里的路由寻径虽然在一定程度上类似传统的导航，但其细节上紧密依赖于专门为无人车导航绘制的高精度地图，所以和传统的导航有本质不同。

路由寻径模块产生的路径信息，直接被下游的行为决策模块所使用。这里的行为决策模块，可以直观地理解成无人车的“副驾驶”。行为决策接收路由寻径的结果，同时也接收感知预测和地图信息。综合这些输入信息，行为决策模块在宏观上决定了无人车如何行驶。这些行为层面的决策包括在道路上的正常跟车，在遇到交通灯和行人时的等待避让，以及在路口和其他车辆的交互通过等。例如，路由寻径要求无人车保持在当前车道（Lane）行驶，当感知到前方有一辆正常行驶的车辆时，行为决策的一个决定便很可能是下达跟车（follow）命令。行为决策模块根据具体实现不同，在宏观上定义的输出指令集合也多种多样。实现行为决策模块的方法相对较多，而且没有非常严格的规则要遵循。实际上，在无人车系统设计中，行为决策模块有时被设计成独立的逻辑模块^{[1][2][3]}，有时其功能在某种程度上和下游的动作规划模块融合到了一起实现^{[4][5][6][7]}。

正是因为行为决策和动作规划需要紧密协调配合，在设计实现两个模块时的一个重要的基本准则是，行为决策模块的输出逻辑需要和下游的动作规划模块逻辑配合一致。动作规划模块，在图 7-1 所示的划分中，解决的是具体的无人车动作（Motion）的规划问题。其功能可以理解为，在一个较小的时空区域内，具体解决无人车从 A 点到 B 点如何行驶的问题。这里动作规划模块需要解决的问题，相对行为决策需要解决的问题，又更加具体了一步。动作规划需要具体把一个短暂时间 t 内从 A 到 B 的中间路径点做出规划，包括选择途经哪些具体的路径点，以及到达每个路径点时，无人车需要达到的速度、朝向、加速度，以及车轮转向等。不仅如此，动作规划还需要保证两点：一是在后续时间内，生成从 A 到 B 的时空路径需要保持一定的一致性；二是这些生成的 A 到 B 之间的路径点，包括到达每个点的速度、朝向、加速度等，在下游反馈控制的车辆和道路的物理属性范围内，是可以实际操作的。

从图 7-1 中可以看到，为了了解所处的周围路况环境并做出行为决策，担当“副驾驶”角色的行为决策模块需要感知和地图定位的输出作为输入（图中实线）。由于行为决策和动作规划模块的紧密联系，一般在系统设计时，我们也会同样让感知和地图定位结果接入动作规划模块。这样相对冗余的设计的好处有两点：一方面，如果仅仅依赖行为决策模块

作为 relay 传递感知结果,那么在行为决策模块计算完成前出现的新感知物体将会被忽略,给无人车的安全带来隐患;二是如果行为决策模块出现了问题,这时的动作规划虽然没有了对交规和四周环境行为层面的决策,但仍然拥有感知和地图完整信息,也能实现最基本的避让,提供无人车的安全性。

规划控制最下层的模块是反馈控制模块。这是一个直接和无人车底层控制接口 CAN-BUS 对接的模块,其核心任务是消化上层动作规划模块的输出轨迹点,通过一系列结合车身属性和外界物理因素的动力学计算,转换成对车辆 Drive-By-Wire 控制的油门、刹车,以及方向盘信号,从而尽可能地控制车去实际执行这些轨迹点。反馈控制模块主要涉及对车辆自身控制,以及和外界物理环境交互的建模。

上述四个模块便是无人车控制规划软件系统的最主要的功能模块。这种模块的划分方法(见参考资料[4]),非常有效地将无人车控制规划这样一个复杂问题,按照计算逻辑从抽象到具体做出了非常合理的切分。这样的划分使得每个模块可以各司其职专注解决本层次的问题,使得复杂软件系统的开发工作,可以实现并行化和模块化,大大提高了开发效率,这是这一划分方法的优势所在。当然随之而来的问题,便是模块之间的协调一致问题,其中最重要的便是模块之间计算结果的一致性问题。本质上,行为决策、动作规划和反馈控制都是在不同层面解决同一个问题。同时他们之间由于上下游关系的存在,其计算结果又互相依赖,所以在具体设计实现各个模块时的一个最重要的准则便是尽可能保证计算结果的一致性和可执行性。行为决策模块在做出决定时,要尽可能保证前后一致且最大可能地让下游动作规划可以执行。动作规划规划的轨迹速度也应当严格在下游反馈控制可以执行的范围内。当冲突出现时,一个普遍的解决冲突的准则是尽可能让上游模块去解决问题迁就下游模块,而不是去 push 推动下游模块的极限。

下面我们就按照图 7-1 中的模块划分,按照从上游到下游的顺序,详细介绍每个模块需要解决的问题。同时我们对于每个模块,结合其需要解决问题的具体场景,详细介绍一到两种常见算法的具体实现,从而使得读者对整套无人车控制规划软件系统的解决方案有一个全面又具体的体验。

7.2 路由寻径

在控制规划模块的最上游是路由寻径模块(也称为寻径模块)。这里的路由寻径和我们常见的如谷歌或者百度的地图导航有着显著不同。普通的谷歌或者百度导航解决的是从

A 点到 B 点的道路层面的路由寻径问题。普通导航其底层导航的元素最小可以具体到某一条路的某一个车道。这些道路和车道都是符合自然的道路划分和标识的。无人车路径规划的 Routing 寻径问题, 虽然也是要解决从 A 点到 B 点的路由问题, 但由于其输出结果并不是为实际的驾驶员所使用, 而是给下游的行为决策和动作规划等模块作为输入, 其路径规划的层次要更加深入到无人车所使用的高精地图的车道 (Lane) 级别。如图 7-2 所示, 其中的箭头线段代表高精地图级别的道路划分和方向。Lane₁, Lane₂, ..., Lane₈ 构成了一条 Routing 输出的路由片段序列。可以看到, 无人车地图级别的 Lane 划分并非和实际的天然道路划分对应。例如, Lane₂, Lane₅, Lane₇ 都代表了由地图定义绘制的“虚拟”转向 Lane。类似地, 一条较长的自然道路也可能被划分为若干个 Lane (例如 Lane₃, Lane₄)。

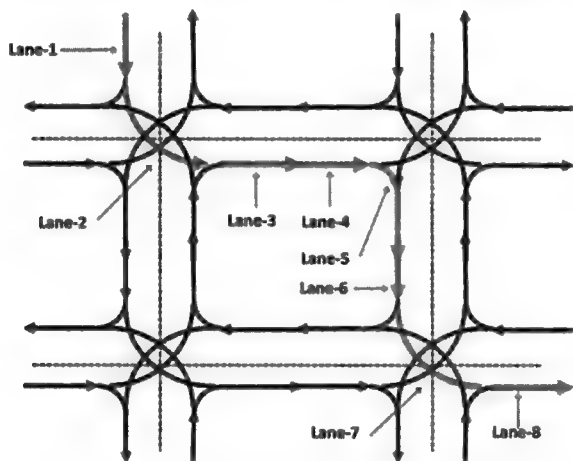


图 7-2 无人车路由寻径模块的高精地图道路级别路由寻径

作为整体无人车控制规划系统的最上游模块, 路由寻径模块的输出严格依赖无人车高精地图 (HD-Map) 的绘制。在高精地图定义绘制的路网 (Road Graph) 的道路 (Lane) 划分的基础上, 以及在一定的最优策略定义下, 路由寻径模块需要解决的问题是计算出一个从起点到终点的最佳道路行驶序列:

$$\{(\text{lane}, \text{start_position}, \text{end_position})\}$$

其中 $(\text{lane}, \text{start_position}, \text{end_position})$ 我们称作一个 Routing Segment (路由片段), 所在的道路由 lane 来标识, $\text{start_position}, \text{end_position}$ 分别代表在这条路由上的起始纵向距离和结束纵向距离。

7.2.1 无人车 Routing 的有向带权图抽象

无人车寻径模块和普通的谷歌或者百度导航不同,无人车 Routing 所考虑的不仅仅局限于路径的长短和拥塞情况等,还需要考虑到无人车执行某些特定行驶动作的难易程度。例如,无人车路由寻径可能会尽量避免在短距离内进行换道,因为无人车的规划控制算法出于安全考虑,需要的换道空间可能比正常的驾驶员驾驶所需要的换道空间更大。从安全第一的原则出发,无人车 Routing 模块可能会给“换道”路径赋予更高的权重 (Cost)。

我们可以把无人车在高精地图的 Lane 级别寻径问题,抽象成一个在有向带权图上的最短路径搜索问题。路由寻径模块首先会基于 Lane 级别的高精度地图,在一定范围内所有可能经过的 Lane 上进行分散“撒点”,我们称这些点为“Lane Point”。这些点代表了对无人车可能经过的 Lane 上的位置的抽样。这些点与点之间,由有向带权的边进行连接,如图 7-3 和图 7-4 所示。一般来说,在不考虑倒车这一特殊情况下, Lane Point 之间是沿着 Lane 行进方向单向可达的关系。连接 Lane Point 之间边的权重,代表了无人车从一个 Lane Point 行驶到另一个点的潜在代价。Lane Point 的采样频率需要保证即使是地图上被分割比较短的 Lane,也能得到充分的采样点。Lane Point 之间的连接具有局部性 (Locality)。自然同一条 Lane 上面的点是前后连接的,但值得注意的是,不同 Lane 之间的 Lane Point 也有相互连接的关系。一个明显的例子是,在转弯时,转弯 Lane 的第一个 Lane Point 和其前驱 Lane 的最后一个 Lane Point 自然连接在一起。另外两条相邻的平行 Lane,在可以合法进行换道的位置 (比如白色虚线位置),其对应位置的 Lane Point 也可能互相连接。图 7-3 给出了几个典型 Lane 连接场景的 Lane Point 之间的权重设置:在任何一个 Lane 的内部采样点 Lane Point 之间,我们把 cost 设置为 1;考虑到右转的代价低于左转,我们把直行接右转的 cost 设置为 5,直行接左转的 cost 设置为 8,右转 Lane 内部 Lane Point 连接 cost 设置为 2,左转 Lane 内部 Lane Point 连接 cost 设置为 3。在图 7-3 所示的换道场景中,两条平行可以换道的 Lane,每条 Lane 内部的连接 cost 依然为 1,但为了突出换道的代价,我们把相邻 Lane 之间的连接权重设置为 10。

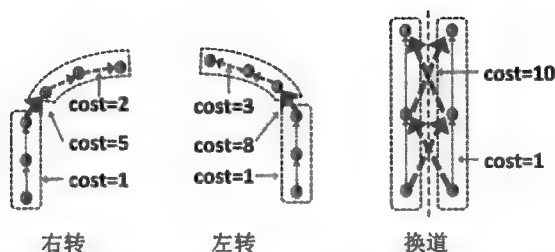


图 7-3 右转、左转和换道 Lane 典型连接场景下 Lane Point 间 cost 的设置

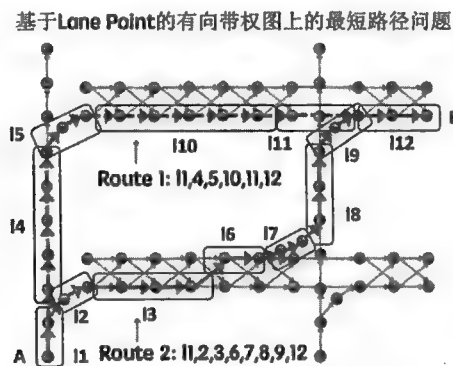


图 7-4 无人车寻径基于 Lane Point 的有向带权图上的最短路径问题抽象

按照图 7-3 设置的 cost，在图 7-4 所示的一个路网（Road Graph）下，我们来对比从 A 到 B 两个可能不同的路由路径 Route 1 和 Route 2。其中 Route 1 对应从 Lane 1 出发，在左下角的路口处直行驶 Lane 4，之后右转（Lane 5），再继续直行经过 Lane 10 和 Lane 11，最后直行经过 Lane 12 到达目的地；Route 2 对应同样从起点 A 所在的 Lane 1 出发，但在左下角的第一个路口处右转接 Lane 2，然后直行并且从 Lane 3 换道至 Lane 6，在右下角路口处经过 Lane 7 左转接直行（Lane 8），最后在右上角的路口处右转（Lane 9）进入最后目的地 B 所在的 Lane 12。即使 Route 2 的实际物理长度小于 Route 1，按照图 7-3 设置的 cost，无人车 Routing 也会偏向于选择总 cost 较小的 Route 2（假设属于不同 Lane 的 Lane Point 之间的连接 cost 除了图 7-3 所示外均为 1，读者可以验证 Route 1 的总 cost 为 22，Route 2 的总 cost 为 44）。

7.2.2 典型无人车路由寻径算法

针对 7.1.1 介绍的无人车 Routing 有向带权图的最短路径问题，我们介绍两种常见的无人车 Routing 算法：Dijkstra 算法^[8]和 A*算法^[9]。

1. Dijkstra 算法在无人车路由寻径中的应用

Dijkstra 算法是一种常见的图论中的最短路径算法，由 Edsger W. Dijkstra 在 1959 年发表。给定一个图中的源节点（Source Node），Dijkstra 算法会寻找该源节点到所有其他节点的最短路径。结合无人车路由的 Lane Point 场景，算法的描述如下。

（1）从高精地图的路网数据接口中读取一定范围的地图 Lane 连接数据，按照 7.2.1 节所述进行 Lane Point 抽样并构建 Lane Point Graph。无人车（也称作 Master Vehicle）所在 Lane 的最接近的 Lane Point 为源节点，目的地所在 Lane 的最接近的 Lane Point 为目的节点。设置源节点到其他所有节点（包括目的节点）的距离为无穷大（inf），源节点到自身的距离为 0。

（2）当前节点设置为源 Lane Point，设置其他所有 Lane Point 为 unvisited（未访问）并且放到一个集合中（Unvisited Set），同时维护一个前驱节点的映射 prev_map，保存每一个 visited 的 Lane Point 到其前驱 Lane Point 的映射。

（3）从当前 Lane Point 节点出发，考虑相邻能够到达的所有未访问的 Lane Point，计算可能的距离（Tentative Distance）。例如，当前 Lane Point X 被标记的距离为 3，Lane Point X 到 Lane Point Y 的距离为 5，那么可能的距离为 $3+5=8$ 。比较该 tentative distance 和 Y 的当前标记距离。如果 Y 的当前标记距离较小，那么保存 Y 的当前标记距离不变，否则更新 Y 的当前标记距离为这个新的 tentative distance 并且更新 prev_map。

（4）对当前 Lane Point 的所有连接的 unvisited Lane Point 重复上述步骤 3 的操作，当所有相连接的 Lane Point 均被操作过之后，标记当前的 Lane Point 为已经 visited，从 unvisited 的集合中去除。已经被 visited 的 Lane Point 的标记距离将不再被更新。

（5）不断从 unvisited 的 Lane Point 集合中选取 Lane Point 作为当前节点并重复步骤 4，直到我们的目标 Lane Point 被从 unvisited 集合中去除；或者在一定范围内的 Lane Point 均已经无法到达（unvisited 集合中最小的 tentative distance 为无穷大，代表从源 Lane Point 无法到达剩下的所有 unvisited Lane Point）。此时，需要返回给下游模块没有可达路径（寻径失败），或者重新读入更大范围的地图路网数据重新开始寻径的过程。

（6）当找到从 A 到 B 的最短路径后，根据 prev_map 进行 Lane 序列重构。

基于 Dijkstra 算法的 Lane Point 有向带权图上的 Routing 算法伪码如图 7-5 所示。其中第 2 到 16 行是典型的 Dijkstra 算法构建每个源 Lane Point 到其他每个 Lane Point 的最小

距离表。之后从第 17 至 22 行，根据得到的每个节点标记的最小距离映射，通过不断查找前驱的 `prev_map` 映射重建最短路径。注意这里的最短路径是一个 Lane Point 的序列，在第 23 行我们对 Lane Point 按照 lane 进行聚类合并最终生成如 `{(lane,start_position,end_position)}` 格式的 Routing 输出。

```

1 function Dijkstra_Routing(LanePointGraph(V,E), src, dst)
2   create vertex set Q
3   create map dist, prev
4   for each lane point v in V:
5     dist[v] = Inf
6     prev[v] = nullptr
7   add v to Q
8   dist[src] = 0
9   while Q is not empty:
10    u = vertex in Q s.t. dist[u] is the minimum
11    remove u from Q
12    for each connected lane point v of u:
13      candidate = dist[u] + cost(u, v)
14      if candidate < dist[v]:
15        dist[v] = candidate
16        prev[v] = u;
17   ret = empty sequence
18   u = dst
19   while prev[u] != nullptr:
20     insert u at the beginning of ret
21     u = prev[u]
22   insert u at the beginning of ret
23   merge lane point in ret with same lane id and return the merged sequence

```

图 7-5 基于 Dijkstra 算法和 Lane Point 网络的路由寻径算法

假设根据 7.2.1 节所描述的 Lane Point 有向带权图生成方法的图有 V 个节点和 E 条边。在使用 minimum priority queue 来优化第 10 行的最小距离查找的情况下，Dijkstra 的 Routing 算法复杂度可以达到 $O(|E| + |V| \log |V|)$ 。

2. A*算法在路径规划上的应用

另一种在无人车路由寻径中常用的算法是 A*算法。A*算是一种启发式的搜索算法。A*算法在某种程度上和广度优先搜索（BFS）、深度优先搜索（DFS）类似，都是按照一定的原则确定如何展开需要搜索的节点树状结构。A*可以认为是一种基于“优点”（best first/merit based）的搜索算法。

A*算法首先会维护一个当前可能需要搜索展开的节点集合（openSet）。每次循环，A*会从这个 openSet 中选取 cost 最小的节点进行展开来继续深入搜索，这个 cost 由 $f(v) = g(v) + h(v)$ 两部分组成。在 A*算法的搜索树结构中，每个节点 v 都有一个由源点到该节点的最小 cost，记为 $g(v)$ ；同时每个节点 v 还对应一个启发式的 cost（称之为 heuristic），

记为 $h(v)$ ；其中 $h(v)$ 作为一个 heuristic，用来估计当前节点 v 到目标节点的最小 cost。当该 heuristic $h(v)$ 满足一定的属性时，A* 能够保证找到源节点到目的节点的最短路径。A* 算法的搜索树在每次循环中都会展开 $f(v) = g(v) + h(v)$ 最小的节点，直到到达目的节点。A* 的详细算法如图 7-6 所示。其中算法第 11 行的 `reconstruct_route` 部分类似于 Dijkstra 算法最后的路由重构部分。`reconstruct_route` 从最后的目的节点出发，通过前驱节点的映射 `prev_map` 向前重构出最终的路由寻径输出。

A* 算法作为一种启发式 (heuristic based) 的搜索算法，当 $h(v)$ 的定义满足 admissible 属性^[9]，即 $h(v, dst)$ 不会超过实际的 $h(v, dst)$ 之间的最小 cost 时，总是能找到最短的路径。当 heuristic 不满足这一条件时，A* 并不能保证找到最短路径。在 7.2.1 节描述的 Lane Point 有向带权图场景下，对于任意两个 Lane Point A 和 B ，一种 heuristic 启发函数的定义为：

$$h(u, v) = \text{dist}(u, v)$$

其中 `dist()` 代表两个 Lane Point 之间在地球经纬度或者墨卡托^[10]坐标系下的距离。

```

1 function AStar_Routing(LanePointGraph(V,E), src, dst)
2   create vertex set closedSet      // set of already visited nodes
3   create vertex set openSet       // set of nodes to be expanded
4   insert src into openSet
5   create map gScore, fScore with default value inf
6   create prev_map with default value nullptr
7   fScore[src] = h(src, dst)
8   while openSet is not empty:
9     current = the node v in openSet s.t. fScore[v] is minimum in openSet
10    if current = dst
11      return reconstruction_route(prev_map, current)
12    remove current from openSet
13    insert current into closedSet
14    for each neighbor u of current:
15      if u is in closedSet:
16        continue; // ignore the neighbor who has already been evaluated
17      candidate_score = gScore[current] + h(current, u)
18      if u not in openSet: // discovered a new node
19        insert u into openSet
20      else if candidate_score <= gScore[u]: // this is not a better path
21        continue;
22      prev[u] = current
23      gScore[u] = candidate_score
24      fScore[u] = gScore[u] + h(u, dst)

```

图 7-6 A* 算法伪码

A* 作为一种最优优先算法 (Best First)，可以看作是 Dijkstra 算法的一种扩展。Dijkstra 算法可以看成是 A* 算法中启发函数 $h(u, v) = 0$ 的一种特例。

7.2.3 路由寻径 cost 设置和强弱路由寻径

在实际的无人车路由寻径计算问题中,往往更重要的不是算法的选择,而是路权(cost)的设置策略。在 7.2.1 节中描述的 cost 调整是整个路由寻径策略的精髓所在,而具体的算法实现(A*或者 Dijkstra)往往并不是最重要的。比如从地图信息我们得知某一条道路的某一条 Lane 非常拥堵,那么可以把进入这条 Lane 上的 Lane Point 之间的连接权重 cost 提高;类似地,如果某条 Lane 被交通管制不能通行,我们也可以相应地把这条 Lane 上的 Lane Point 设置为互相不可达,从而使得算法不会去选择某条特定的 Lane。路由寻径的 Lane Point 之间的 cost 可以根据不同策略实时灵活调整,为无人车路由寻径提供支持。考虑到实际的路网数据往往较大,基于 Lane Point 有向带权图的最短路径往往是在提前 pre load 的部分地图路网数据上进行的。如果出现在较小范围内不可达的情况,可能需要重新读入更大的路网和地图数据进行重新路由寻径。

对路由寻径模块产生路由计算的请求,有两种情况:一是当无人车开始行驶时,由用户来设置起点到终点,从而触发路由寻径请求;另一种情况中,请求是由下游模块发起的。这里我们讨论“强 Routing”和“弱 Routing”两种系统设计。“强 Routing”指的是下游模块(如 Behavior Decision 及 Motion Planning)等严格遵守路由寻径模块的输出。例如路由寻径模块要求按照某条 Lane X 行驶,但感知发现 Lane X 上有一个非常慢的障碍车辆,在强路由的设计下,无人车会严格执行在 Lane X 上行驶;但在“弱 Routing”的设计下,无人车可能会短暂跨越到相邻的 Lane,超过障碍车辆,再回到 Lane X 继续行驶。无论是“强 Routing”还是“弱 Routing”,当出现需要紧急避让,或者周围交通情况导致无人车无法执行当前的路由寻径结果时,无人车会按照安全第一的原则继续行驶,并且发起重新路由寻径的请求。

7.3 行为决策

行为决策层在整个无人车规划控制软件系统中扮演着“副驾驶”的角色。这个层面汇集了所有重要的车辆周边信息,不仅包括了无人车本身的当前位置、速度、朝向,以及所处车道,还收集了无人车一定距离以内所有重要的感知相关的障碍物信息。行为决策层需要解决的问题,就是在知晓这些信息的基础上,决定无人车的行驶策略。这些信息具体包括以下几点。

(1) 所有的路由寻径结果:比如无人车为了到达目的地,需要进入的车道是什么

(target lane)。

(2) 无人车的当前自身状态: 车的位置、速度、朝向, 以及当前主车所在的车道、按照寻径路由需要进入的下一个车道等。

(3) 无人车的历史信息: 在上一个行为决策周期, 无人车所做出的决策是什么? 是跟车、停车、转弯或者是换道?

(4) 无人车周边的障碍物信息: 无人车周边一定距离范围内的所有障碍物信息。例如周边的车辆所在的车道, 邻近的路口有哪些车辆, 它们的速度、位置如何, 以及在一个较短的时间内它们的意图和预测的轨迹, 周边是否有自行车或者行人, 以及他们的位置、速度、轨迹等。

(5) 无人车周边的交通标识信息: 一定范围内的 Lane 的变化情况。比如路由寻径的结果是在 Lane 1 的纵向位移 10m 处换道进入对应的相邻 Lane 2 的纵向位移 20m 处, 那么 Lane 1 的合法的纵向位移换道空间是多大? 比如从一个直行 Lane 行驶结束, 需要进入下一个左转 Lane, 两条 Lane 的交界处是否有红绿灯或者人行道?

(6) 当地的交通规则: 例如道路限速, 是否可以红灯右拐等。

无人车的行为决策模块, 就是要在上述所有信息的基础上, 做出如何行驶的决策。可以看出, 无人车的行为决策模块是一个信息汇聚的地方。由于需要考虑如此多种不同类型的信息及受到非常本地化的交规限制, 行为决策问题往往很难用一个单纯的数学模型解决。往往更适合行为决策模块的解决方法, 是利用一些软件工程的先进观念来设计一些规则系统。例如在 DARPA 无人车竞赛中, Stanford 的无人车系统“Junior”利用一系列 cost 设计和有限状态机 (Finite State Machine) 来设计无人车的轨迹和操控指令。类似地, CMU 的无人车系统“Boss”^{[18][19]}通过计算分析 Lane 之间的空隙 (Gap), 并且按照一定规则和一些预设的阈值比较决定换道这一行为的触发。其他很多的参赛系统如 Odin 和 Virginia Tech 也都利用了规则引擎来决定无人车的驾驶行为。Carolo 团队则是结合了规则引擎和行为模型, 建立了一个混合的无人车决策系统。随着对无人车研究兴趣的广泛发展和研究的深入, 越来越多的研究结果^[9]开始使用一些 Bayesian 模型对无人车行为进行建模。其中 MDP (Markov Decision Process) 和 POMDP (Partially Observable Markov Decision Process) 都是在学术界最为流行的无人车行为决策建模方法, 我们将在本节简单介绍几种基于 MDP 的无人车行为决策方式。虽然 MDP 类的非 deterministic 概率模型在学术界渐渐流行, 但笔者从工业界的实际应用经验出发, 认为基于规则的决定性 (Deterministic) 行为决策系

统仍然是目前工业界的主流。本节将介绍一种利用分治 (Divide and Conquer) 思想来设计的基于规则的行为决策实现。事实上,如果能够用先进的软件工程实现结合交规和周边路况的行为决策,作者认为 **Deterministic** 的规则系统甚至可能在安全可靠上优于基于概率模型的实现方式。设想实际人类驾驶员是如何按照一个固定的路线从 A 点开到 B 点。因为交通规则是明确并且是可以具体执行的,所以作者认为在宏观层面的驾驶行为,在给定的周边路况下,按照交规要求和自身的意图,可以看成是完全基于规则的决定性行为。

7.3.1 有限状态马尔可夫决策过程

一个马尔可夫决策过程,由下面的五元组定义: (S, A, P_a, R_a, γ) 。

(1) S 代表了无人车所处的有限的状态空间,状态空间的划分可以结合无人车当前位置及其在地图上的场景进行设计:例如在位置维度可以考虑将无人车按照当前所处的位置划分成等距离的格子;参考地图的场景,可以将无人车所处的车道和周边道路情况归纳到有限的抽象状态中。

(2) A 代表了无人车的行为决策空间,即无人车在任何状态下的所有行为空间的集合:例如,可能的状态空间包括当前 Lane 跟车 (Follow)、换道 (Change Lane)、左/右转 (Turn Left/Right)、路口的先后关系 (Yield/Overtake)、遇到行人或者红绿灯时的停车 (Stop) 等。

(3) $P_a(s, s') = P(s' | s, a)$ 是一个条件概率,代表了无人车在状态 s 和动作 a 下,到达下一个状态 s' 的概率。

(4) $R_a(s, s')$ 是一个激励函数 (Reward),代表了无人车在动作 a 下,从状态 s 到状态 s' 所得到的激励。该激励函数的设计可以考虑安全性、舒适性,以及下游动作规划 (Motion Planning) 执行难度等因素综合设计。

(5) $\gamma \in (0, 1)$ 是激励的衰减因子,下一个时刻的激励便按照这个因子进行衰减;在任何一个时间,当前的激励系数为 1,下一个时刻的激励系数为 γ ,下两个时刻的激励系数为 γ^2 ,依此类推。其含义是当前的激励总是比未来的激励重要。

无人车行为决策层面需要解决的问题,在上述 MDP 的定义下,可以正式描述为寻找一个最优“策略”,记为 $\pi: S \rightarrow A$ 。在任意给定的状态 s 下,策略会决定产生一个对应的行为 $a = \pi(s)$ 。当策略确定后,整个 MDP 的行为可以看成是一个马尔可夫链。行为决策的策略 π 的选取目标是优化从当前时间点开始到未来的累积激励 (如果 Reward 是随机变量,则优化累积 Reward 的期望):

$$\sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1}), \text{ 其中 action 是由策略 } \pi \text{ 产生 } a = \pi(s)$$

在上述马尔可夫决策过程定义下, 最优策略 π 通常可以用动态编程 (Dynamic Programming) 的方法求解。假设转移矩阵 P 和激励分布 R 已知, 最优策略的求解通常都是基于不断计算和存储如下两个基于状态 s 的数组:

$$\pi(s_t) \leftarrow \operatorname{argmax}_a \left\{ \sum_{s_{t+1}} P_a(s_t, s_{t+1}) (R_a(s_t, s_{t+1}) + \gamma V(s_{t+1})) \right\},$$

$$V(s_t) \leftarrow \sum_{s_{t+1}} P_{\pi(s_t)}(s_t, s_{t+1}) (R_{\pi(s_t)}(s_t, s_{t+1}) + \gamma V(s_{t+1}))$$

其中数组 $V(s_t)$ 代表了未来衰减叠加的累积 (期望) 激励, $\pi(s_t)$ 代表需要求解的策略。具体的求解过程可以是在所有可能的状态 s 和 s' 之间进行重复迭代计算, 直到二者收敛为止。更进一步, 在 Bellman 的 Value Iteration 算法中, $\pi(s_t)$ 不需要进行显式的计算, 而是可以将其必要的计算包括在 $V(s_t)$ 的计算中, 因此可以得到如下的 Value Iteration 的单步迭代计算:

$$V_{i+1}(s) \leftarrow \max_a \left\{ \sum_{s'} P_a(s, s') (R_a(s, s') + \gamma V_i(s')) \right\}$$

其中 i 代表迭代步骤, 在 $i=0$ 时使用一个初始猜测 $V_0(s)$ 开始迭代, 直到 $V(s)$ 的计算趋于稳定为止。由于利用 MDP 建模解决无人车行为决策的方法比较多样, 本书在这里不再赘述所有的基于马尔可夫决策过程的行为决策方法, 读者可以参考参考文献中的[2][3]来了解具体的状态空间、动作空间, 以及转移概率和 Reward 函数的实现举例。需要强调的是, 利用 MDP 解决无人车行为决策的最关键部分在于激励函数 R 的设计。在设计这一 Reward 函数时, 需要尽可能考虑如下因素。

(1) 到达目的地: “鼓励” 无人车按照既定的路由寻径路线行进到达目的地, 也就是说, 如果选择的动作 $a = \pi(s)$ 会使得无人车有可能偏离既定的路由寻径路线, 那么应当给予对应的惩罚。

(2) 安全性和避免碰撞: 按照前文所述, 如果将无人车周边的空间划分成等间距的方格, 那么远离可能有碰撞的方格应当得到奖励, 接近碰撞发生时, 应当加大惩罚。

(3) 乘坐的舒适性和下游执行的平滑性 (smoothness): 这两个因素往往是一致的。

乘坐的舒适往往意味着安全顺畅的操作。例如从某一个速度状态到一个比较接近的速度状态的 $a = \pi(s)$ ，其 cost 应该较小；反之，如果猛打方向盘或者猛然加速，这个 $a = \pi(s)$ 的 action ，对应的 cost 就应该比较高（负向 Reward ）。

正是因为利用马尔可夫概率模型的 MDP 需要如此细致地设计诸如状态空间、转移概率和激励函数等参数，作者认为基于规则的宏观行为决策系统是一种更可靠的设计，下面我们介绍一种利用场景分割无人车周边环境，通过构建子场景并加以 Rule 运用的行为决策系统设计。

7.3.2 基于场景划分和规则的行为决策设计

这里我们介绍一种基于规则的无人车行为决策层的设计，其核心思想是利用分治的原则将无人车周边的场景进行划分。在每个场景中，独立运用对应的规则来计算无人车对每个场景中元素的决策行为，再将所有划分的场景的决策进行综合，得出一个最后综合的总体行为决定。我们先引入几个重要概念：综合行为决策（**Synthetic Decision**）、个体行为决策（**Individual Decision**），以及场景（**Scenario**）。

1. 综合决策

综合的行为决策代表无人车行为决策层面的整体最高层的决策，例如按照当前 Lane 跟车保持车距行驶，换道至左/右相邻 Lane，立刻停车到某一停止线后等；作为最高层面的综合决策，其所决策的指令状态空间定义，需要和下游的动作规划（**Motion Planning**）协商一致，使得做出的综合决策指令是下游可以直接用来执行规划出路线轨迹（**trajectory**）的。为了便于下游直接执行，综合决策的指令集往往带有具体的指令参数数据。表 7-1 中列出了一些综合决策的指令集定义及其可能的参数数据。例如，当综合决策是在当前车道跟车行驶（**Follow**）时，传给下游动作规划的不仅是跟车这一宏观指令，还包含如下参数数据：前方需要跟车的车辆的 id （一般从感知输出获得），跟车需要保持的车速（当前车道限速和前车速之间较小值），以及需要和前车保持的距离（例如前车尾部向后 3m）等。下游的动作规划基于宏观综合决定及伴随指令传来的参数数据，结合地图信息（如车道形状）等，便可以直接规划出安全无碰撞的行驶路线。

表 7-1 行为决策中的综合决策及其参数

综合决策	参数
行驶	当前车道 (Current lane) 目的车速 (Speed limit of the current lane)
跟车	当前车道 (Current lane) 跟车对象 (Id for the vehicle in front to follow) 目的车速 (Speed to reach minimum of current lane speed limit and speed of the vehicle to follow) 跟车距离 (not exceeding 3m behind the vehicle in front)
转弯	当前车道 (Current lane) 目的车道 (Target lane) 转弯属性 (Left or right turn) 转弯速度 (Speed limit for turning)
换道	当前车道 (Current lane) 换道车道 (Target lane) 加速并道 (Change lane by overtaking and speed up to 10m/s) 减速并道 (Change lane by yielding and speed down to 2m/s)
停车	当前车道 (Current lane) 停车对象 (Id for any object to stop, if any) 停车位置 (Stop by 1m behind the object to stop)

2. 个体决策

与综合决策相对应的是个体决策。在本节开始处我们便提出过，行为决策层面是所有信息汇聚的地方。因此，最终的综合决策必须是考虑了所有重要的信息元素后得出的。这里，我们提出对所有重要的行为决策层面的输入个体，都产生一个个体决策。这里的个体，可以是感知输出的路上车辆和行人，也可以是结合了地图元素的抽象个体，比如红绿灯或者人行横道对应的停止线等。事实上，最终的综合决策是先经过场景的划分，产生每个场景下的个体决策，再综合考虑归纳这些个体决策才得到最终的综合决策。个体决策和综合决策相似的地方是除了其指令集本身外，个体决策也带有参数数据。个体决策不仅是产生最后的综合决策的元素，而且也 和综合决策一起被传递给下游动作规划模块。这种设计虽然传递了更多的数据，但作者根据工业界的经验认为，传递作为底层决策元素的个体决策能够非常有效地帮助下层模块更有效地实现路径规划。同时，当需要调试解决问题时，传递过来的个体决策能够大大提高调试的效率。表 7-2 列出了一些典型的个体决策及其可能的附带参数数据。例如，在做出针对某个感知物体 X 的超车这一个体决策时，附带的参

数数据包括超车的距离和时间限制。距离代表本车车身至少要超过物体 X 的车头的最小距离,同样,时间代表这段超车安全距离至少要对对应物体 X 行驶一个最小安全时间间隔。注意这种超车个体决策,往往发生在两车轨迹有所交互的场景中。典型的场景包括换道和路口的先后后行。下面我们会结合红绿灯路口右转这一具体例子,描述如何结合分割场景产生不同的个体决策,并最终融合成综合决策输出。

表 7-2 行为决策中的个体决策及其参数

个体决策		参数
车辆	跟车	跟车对象 (Id for the vehicle to follow) 跟车速度 (Speed to reach for following the vehicle) 跟车距离 (Distance to keep for following the vehicle)
	停车	停车对象 (Id for the vehicle to stop) 停车距离 (Distance to stop behind the vehicle)
	超车	超车对象 (Id for the vehicle to overtake) 超车距离 (Minimum distance to keep for overtaking) 超车时间 (Minimum time gap to keep for overtaking)
	让行	让行对象 (Id for the vehicle to yield) 让行距离 (Minimum distance to keep for yielding) 让行时间 (Minimum time gap to keep for yielding)
行人	停车	停车对象 (Id for the pedestrian to stop) 停车距离 (Minimum distance to stop by the pedestrian)
	躲避	躲避对象 (Id for the pedestrian to swerve) 躲避距离 (Minimum distance to keep while swerving around)

3. 场景划分构建和系统设计

个体决策的产生依赖于场景的构建。这里我们可以将场景理解成一系列具有相对独立意义的无人车周边环境的划分。利用这种分而治之思想的场景划分,我们将无人车行为决策层面汇聚的众多无人车主车周边属于不同类别的信息元素,聚类到不同的富有实际意义的场景实体中。在每个场景实体中,我们通过交规,并结合主车的意图,计算出对于每个信息元素的个体决策,再通过一系列准则和必要的运算把这些个体决策最终综合输出给下游。

图 7.7(a)和图 7.7(b)所示为两个非常典型的场景划分。

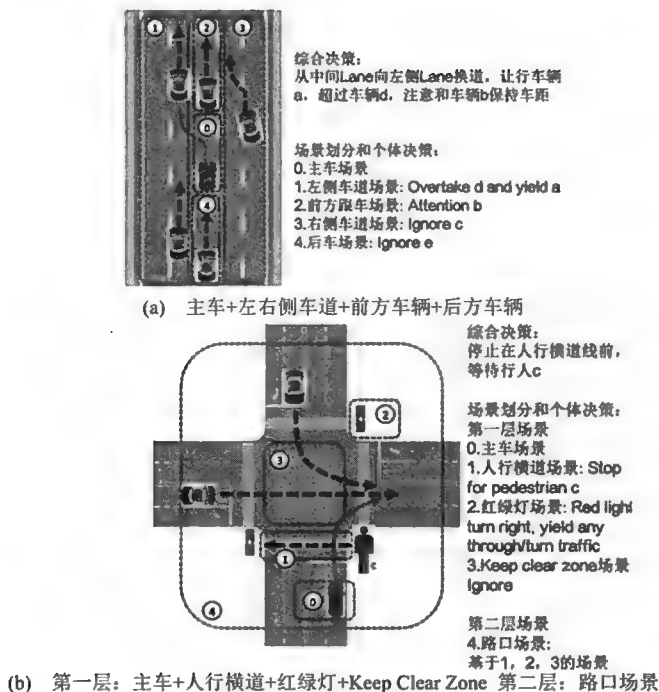


图 7-7 行为决策场景

在图 7.7(a)中，车辆 a 和 d 出现在“左侧车道”这一场景①中。此时无人车主车的意图是向左换道。在计算了主车相对 a 和 d 的位置和速度后，“左侧车道”这一场景计算的结果是需要让 a 车先通过，然后在 d 车之前进行换道；与此同时，一个相对独立的场景是“前车”场景②，此时主车虽然在考虑向左换道，但仍然需要注意当前车道的前车，所以场景②对前车仍然做出了对车辆 b 需要注意这个个体决策；相对主车当前意图而言，右侧车道场景③和后方车辆场景④和当前的主车轨迹没有冲突，所以可以安全做出对车辆 c 和 e 的 Ignore 决策。

值得一提的是类似前方后方车辆，两侧车道这些场景是基本的场景。有一些场景的基本元素本身就可以是这些基本场景。图 7.7(b)中给出了“路口”这么一个“复合场景”。可以看出，我们的场景定义是分层次的 (Layered)。每个层次中间的场景是互相独立构建的。其中主车可以认为是最基本的底层场景，其他所有场景的构建都需要先以无人车主车在哪里这么一个基本场景为基础；在此之上的第一层场景包括红绿灯、前后方车辆，以及左右两侧车道车辆等；如图 7.7(b)所示中的路口场景，是第二层的复合场景。其中的元素包括第一层的人行横道、红绿灯，以及主车等场景。结合这些场景，路口场景本身中的元

素是车辆 a 和 b。假设此时无人车的意图是右转，路口红灯可以右转但由于没有道路优先权需要避让其他车辆，此时感知发现一个行人在人行横道的场景横穿马路，那么结合所有这些场景元素和意图，得到的最终指令是针对行人在人行横道前停车。

综上所述，每个场景模块利用自身的业务逻辑（Business Logic）来计算其不同元素个体的决策。通过场景的复合，以及最后对所有个体的综合决策考虑，无人车得到的最终行为决策需要是最安全的决策。这里的一个问题是会不会出现不同场景对同一个物体（例如某个车辆）通过各自独立的规则计算出矛盾的决策？从场景的划分可以看出，本身一个物体出现在不同场景里的概率是很小的。事实上，我们提出的这种场景划分的方法本身就尽可能避免了这一情况的出现。即使这种矛盾出现，在图 7-8 所示的系统框架的中间层，也会对所有的个体决策进行汇总和安全无碰撞的验证。

整个行为决策层面的框架和运行流程如图 7-8 所示。首先是结合主车信息、地图数据及感知结果构建不同层次的场景。在路由寻径的指引下，每个场景结合自身的规则（往往是交规或者安全避让优先），计算出属于每个场景物体的个体决策。在所有的个体决策计算完毕后，虽然发生的概率极其微小，但我们还是会检查有无冲突的个体决策。在对冲突的个体决策进行冲突解决（往往是优先避让）后，我们会在一个统一的时空里，推演预测当前的所有个体决策能否汇总成一个安全行驶无碰撞的综合决策。如果这样的安全无碰撞综合决策存在，我们便将其和个体决策一起输出给下层的动作规划模块计算具体从当前位置到下一个位置的时空轨迹。

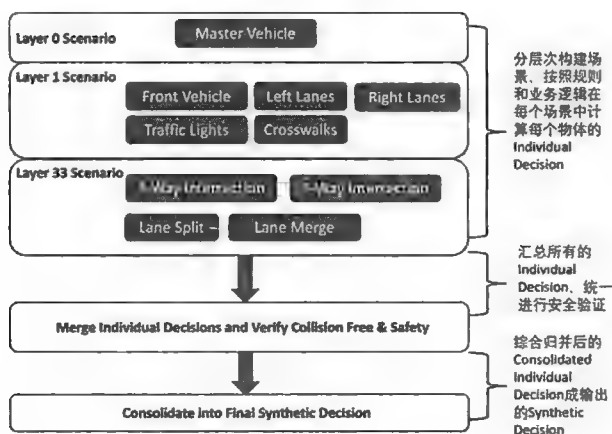


图 7-8 基于规则和场景的行为决策模块系统框架和流程

7.4 动作规划

在行为决策层下游的模块是动作规划,其任务是将行为决策的宏观指令解释成一条带有时间信息的轨迹曲线,给底层的反馈控制进行实际对车的操作。这里无人车的动作规划,可以看作是普通机器人动作规划(Robotic Motion Planning)的一种特殊场景。事实上,作者认为无人车的动作规划问题是整个机器人动作规划领域里相对简单的一个问题。这是因为车辆的轨迹是依附于一个二维平面的。车辆在方向盘、油门的操控下,其行驶轨迹的物理模型相对于普通的机器人姿态的动作轨迹更简单。从 DARPA 无人车比赛开始,无人车动作规划便逐渐成为一个相对独立的模块^[10],尝试在城市道路行驶及停车等综合条件下解决路径规划的问题,也有一些在特定场景下的路径规划问题的解决方法。参考资料中的[4]和[11]列出了近年来动作规划的很多不同方向的工作,读者可以作为参考。随着这些研究的进展,路径模块需要解决的问题也逐渐明晰:几乎所有动作规划都试图解决在一定的约束条件下优化某个范围内的时空路径问题。这里所谓的“时空路径”指车辆在一定时间段行驶的轨迹。该轨迹不仅包括位置信息,还包括整条轨迹的时间信息和车辆姿态:即到达每个位置的时间、速度,以及任何可能的和时间相关的运动变量如加速度、曲率、曲率的高阶导数等信息。由于车辆控制是一个不和谐的系统^[12],车辆的实际运行轨迹总是呈现出属于平滑的类似螺旋线的曲线簇的属性。因此,轨迹规划这一层面需要解决的问题,往往可以非常好地抽象成一个在二维平面上的时空曲线优化问题。考虑动作规划这个层面的优化问题所需要的两个要素:一是需要优化的函数(Object)/代价(Cost)目标;二是边界条件的限制(Constraint)。结合图 7-1 所示的整个系统框架,这里的优化目标函数,往往以 Cost 函数的形式呈现,优化的目标是找到满足边界条件限制的最小 Cost 的曲线。这里的 Cost 和如下几个重要因素紧密相关。首先是上游的行为决策输出的决策结果。作为下游直接规划无人车路线曲线的动作规划,其优化目的必须满足达到行为层面的要求。这些要求往往体现在曲线的长度不能超过某一停止线,曲线横向位移不能触碰到需要避让的物体等;其次,由于我们着重考虑在城市综合道路(Urban Road)上的行驶,车辆行驶的曲线要考虑和道路的关系,即动作规划的曲线要满足能够沿道路行驶的基本要求,这些要求也会被转化成曲线的不同代价来体现;在动作规划的边界条件限制层,往往需要更多考虑的是如图 7-1 所示的下游反馈控制模块。例如,车辆的转向由方向盘控制导致车辆的曲率和曲率二阶导变化受到一定的限制。车辆的油门加速同样限制车辆的加速度的变化率不可能过大等。

这里我们借鉴参考资料[11]中的动作规划算法,提出一种更简单明确地将动作规划问

题拆分成两个问题——轨迹规划 (Trajectory Planning) 和速度规划 (Speed Planning) 来解决的思路。其中轨迹规划只解决在二维平面上, 根据行为决策和地图信息定义的 Cost 函数下优化轨迹的问题。这里的轨迹不考虑速度因素, 只是单纯的不同长度的轨迹曲线; 而速度规划问题则是在选定了一个或者若干个轨迹 (Trajectory) 之后, 解决用什么样的速度行驶这条轨迹的问题。相比于参考资料[11]中的联合优化带有时间和速度信息的时空轨迹, 这样的方法使得每个层次定义的问题更加清晰和相对易于建模解决。虽然分开优化不一定能保证达到联合意义上的最优解, 但是在实际的工程实践中, 作者认为分开优化是更实际有效的解决方案。下面我们就分别详细介绍轨迹规划和速度规划的算法。

7.4.1 轨迹规划

1. 车辆模型、道路定义, 以及候选轨迹生成

我们首先介绍车辆和道路的数学模型。对于车辆, 我们考虑车辆的姿态向量 $\bar{x} = (x, y, \theta, \kappa, v)$, 其中 (x, y) 表示车辆在二维平面的位置, θ 表示车辆的朝向, κ 表示曲率 (即朝向 θ 的变化率), v 表示车辆的速度 (即轨迹任意点的切线速度)。车辆的这些姿态变量的标量大小满足如下关系:

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= v \kappa\end{aligned}$$

其中曲率 κ 的大小往往由系统的输入限制条件决定。在此基础上, 考虑一条由车辆运动产生的连续轨迹 (Path)。我们称沿着轨迹的方向的位移为 S 方向。轨迹相对于车辆姿态的系统关系由下列偏微分方程式给出:

$$\begin{aligned}dx / ds &= \cos(\theta(s)) \\ dy / ds &= \sin(\theta(s)), \\ d\theta / ds &= \kappa(s)\end{aligned}$$

注意这里我们并没有对 κ 和 θ 之间的关系做出特定限制, 即车辆可以在任意朝向 θ 上任意改变其曲率 κ 。在实际的车辆模型中, 车辆的曲率 κ 和朝向 θ 之间是有一定限制的, 但这个微小的模型偏差并不影响我们这个动作规划算法的一般性和实用性。

我们的轨迹规划 (Trajectory Planning) 算法非常依赖于地图中对于道路中心线 (Center Line) 的定义。这里我们认为道路是由道路的采样函数所定义。采样函数为: $r(s) = [r_x(s), r_y(s), r_\theta(s), r_\kappa(s)]$, 其中 s 代表道路的中心线切向方向的位移 (以后也称为纵向

位移 s)。与此对应的是道路的中心线垂直方向位移 l , 也称为横向位移。如果考虑一个车辆的姿态点具体关系, 其各个分量由道路坐标系下的 (s, l) 坐标, 以及道路采样函数 (s, l) 决定, 满足:

$$x_r(s, l) = r_x(s) + l \cos(r_\theta(s) + \pi/2)$$

$$y_r(s, l) = r_y(s) + l \sin(r_\theta(s) + \pi/2)$$

$$\theta_r(s, l) = r_\theta(s)$$

$$\kappa_r(s, l) = (r_\kappa(s)^{-1} - l)^{-1}$$

其中曲率 κ_r 在道路转弯的内侧加大 (随纵向位移 l 加大), 外侧则减小。我们使用右手坐标系, 所以如图 7-9 所示, 在靠近原点处朝 x 轴的正方向, 纵向位移 l 朝着 y 轴正方向加大。假设对于某条道路 Lane(k), 其纵向宽度 l_k 保持不变。那么该条道路可以表示为一个随着中心线横向位移 s 的点集 $\{p(s, l_k) : s \in \mathbf{R}^+\}$ 。我们称这样的一个坐标系统为 SL 坐标系统。

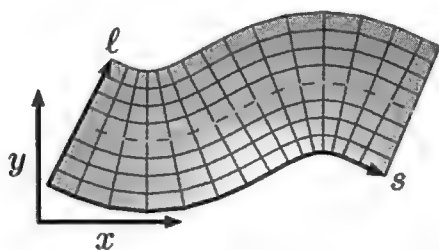


图 7-9 XY 平面下的 SL 坐标系统及其网格划分

在上述的车辆模型和道路模型下, 我们讨论轨迹规划所产生的轨迹曲线。首先, 我们定义车辆的轨迹 (Trajectory) 为一个从 $[0, 1]$ 区间到车辆姿态向量集合 $C = \{\bar{x}\}$ 的连续映射: $\rho: [0, 1] \rightarrow C$ 。其中车辆的初始姿态向量为 $\bar{x} = (x, y, \theta, \kappa)$ 。每条轨迹终点处如图 7-10 所示, 轨迹 1 的终点姿态为 $\rho_1(1) = q_{\text{end}1}$, 轨迹 2 的终点姿态向量为 $\rho_2(1) = q_{\text{end}2}$, 初始姿态为 $\rho_1(0) = \rho_2(0) = q_{\text{init}}$ 。轨迹优化的目标便是在所有可能的轨迹曲线中, 筛选出满足边界条件的轨迹曲线, 再寻找一条/若干条最平滑且 Cost 函数最低的曲线。其中轨迹的候选曲线我们用类似在路由寻径模块中介绍的“撒点”的采样方式生成。参考图 7-10 在某条 Lane 的 SL 坐标系下, 均匀切分的 S 和 L 方向的方格内, 在固定 S 和 L 间隔下, 考虑每个 (s_i, l_j) 区域的中心点。一条候选的轨迹便可以看作是沿着 Lane 的中心线纵向位移 s 方向连接不同 Trajectory Point 的平滑曲线。在图 7-10 所示的道路 SL 分割和采样下, 可能的 Trajectory Point 有 16 个 (4 个 s 位置, 4 个 l 位置), 从车辆的初始位置出发, 我们只考虑在 s 方向

单调增大的可能，不考虑城市综合道路行驶中的倒车情况，那么总的候选曲线的条数为 $4^4 = 256$ 条。轨迹优化便是要在这 256 条候选的曲线中找出最平滑且 Cost 最优的轨迹。

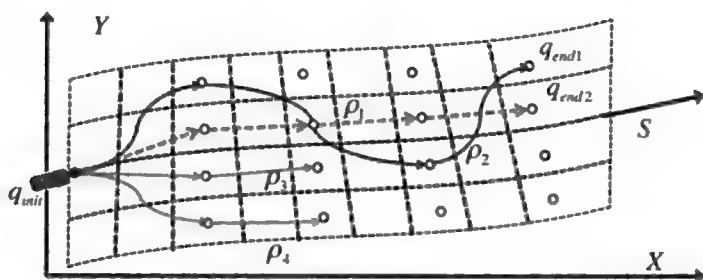


图 7-10 SL 坐标系下道路的分割采样及可能的轨迹

我们采用多项式螺旋线^[13]连接轨迹点，从而生成候选的曲线。多项式螺旋线，如图 7-11 所示，代表了一类曲率可以用弧长（对应我们轨迹中的 s 方向）的多项式函数来表示的曲线簇。我们使用三阶（Cubic）或者五阶（Quintic）的多项式螺旋线，其曲率 k 和轨迹弧长 s 的关系 $k(s)$ 为：

$$k(s) = k_0 + k_1s + k_2s^2 + k_3s^3, \text{ 或者}$$

$$k(s) = k_0 + k_1s + k_2s^2 + k_3s^3 + k_4s^4 + k_5s^5$$

这里我们使用三阶或者五阶的多项式螺旋线拟合，是遵循常见的已有动作规划方面工作的惯例。曲线的阶数对于其在曲线连续性方面并没有本质的影响，其他阶数（如四阶甚至更高阶）的多项式曲线簇也能够作为轨迹点之间的连接曲线。三阶和五阶多项式在满足边界条件约束上有一个重要的区别：三阶多项式螺旋线会导致曲率的二阶导 d^2k/ds^2 （对应方向盘转速）的不连续，而五阶多项式则可以同时保证 dk/ds 和 d^2k/ds^2 的连续性。三阶和五阶多项式螺旋线在速度较低时，该差别在反馈控制上引入的误差体现并不明显，但在速度较快时该误差不可忽略。

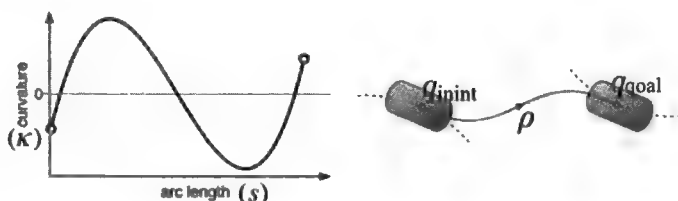


图 7-11 多项式螺旋线及车辆姿态的螺旋线示意图

基于这种使用三阶（五阶）螺旋线连接的轨迹，其参数可以快速有效地通过梯度下降（Gradient Descent）的方法来搜索。以三阶多项式为例，我们考虑从车辆初始姿态 $q_{\text{init}} = (x_I, y_I, \theta_I, \kappa_I)$ 到目标姿态 $q_{\text{goal}} = (x_G, y_G, \theta_G, \kappa_G)$ ，且具有连续曲率的三阶螺旋线： $k(s) = k_0 + k_1s + k_2s^2 + k_3s^3$ 。在初始状态 $s=0$ 时，考虑曲率的一阶导数和二阶导数均需要满足初始状态的限制，我们可以得到：

$$\begin{aligned}\kappa_0 &= \kappa_I \\ \kappa_1 &= d\kappa(0)/ds \\ \kappa_2 &= d^2\kappa(0)/ds^2\end{aligned}$$

这样使得实际未知参数减少到 2 个（ κ_3, s_G ），利用梯度向量我们可以快速寻找到非常接近初始状态限制的三阶螺旋线的参数。

2. 基于轨迹点的有向图构建和搜索

在之前描述的车辆道路模型、轨迹点以及连接轨迹点的多项式螺旋线等的设定之下，轨迹规划简化成针对 $|l_{\text{total}}/\Delta l| \times |s_{\text{total}}/\Delta s|$ 个轨迹点连接成的 $|l_{\text{total}}/\Delta l|^{s_{\text{total}}/\Delta s}$ 条潜在候选曲线的搜索问题。考虑这些轨迹点构成的图 $G=(V,E)$ 。其中每个轨迹点都是图中的一个节点， $v \in V, v=(x,y,s,l)$ ；对于任意两个节点 $v,u \in V$ ，当其对应的 s 坐标满足 $s_v < s_u$ 时， $e(v,u) \in E$ 代表从 x 到 y 的三阶/五阶多项式螺旋曲线。最优的曲线搜索问题转化为在上述有向带权图上的“最短路径”搜索问题。需要注意，这里的最短路径较为特殊的是其不仅包含沿着路径的累积 Cost，还包括了当路径轨迹确定后这条路径的整体 Cost。考虑由轨迹点 n_0, n_1, \dots, n_k 连接成的曲线 τ ，其中初始轨迹点为 n_0 ，轨迹终点为 n_k ，那么该轨迹的 Cost 可以写成：

$$\Omega(\tau) = c(\tau) + \Phi(\tau)$$

其中 $c(\tau)$ 代表沿该曲线行驶累积的 Cost， $\Phi(\tau)$ 代表这条曲线在此终点终结而引入的 Cost。如果将 $\Phi(\tau)$ 函数写成按照轨迹点的增量形式，那么：

$$\Omega(\tau(n_0, n_1, \dots, n_k)) = g(n_k) + \Phi_c(\tau(n_{k-1}, n_k))$$

其中我们定义函数 $g(n)$ 将表示“到达”节点 n 的最小 Cost，注意该 Cost 包含沿途轨迹螺旋线的 Cost 累积而并不包括以 n 为终点而引起的整体轨迹曲线 Cost 增长。那么，考虑以 n_{k-1} 为倒数第二个节点的所有轨迹中，最小 Cost 轨迹曲线的最后一个轨迹点 n_k 的选取。该节点 n_k 需要满足：

(1) 存在节点 n_{k-1} 到节点 n_k 的有向边 $e(n_{k-1}, n_k)$ ，也记为 $\tau(n_{k-1}, n_k)$ 。

(2) 对所有节点 n_{k-1} 能够到达的节点集合 $\{\tilde{n}_k\}$ (即存在有向边 $e(n_{k-1}, \tilde{n}_k)$) 中, 以 n_k 结束的轨迹曲线的总 Cost 最小。

$$n_k \leftarrow \underset{\tilde{n}_k}{\operatorname{argmin}} g(n_{k-1}) + c(\tau(n_{k-1}, \tilde{n}_k)) + \Phi_c(\tau(n_{k-1}, \tilde{n}_k)),$$

这样我们可以更新 $g(n_{k-1})$ 为: $g(n_k) \leftarrow g(n_{k-1}) + c(e(n_{k-1}, n_k))$ 。

我们可以用图 7-12 所示的动态编程算法来计算从车辆的初始位置 n_0 节点起始, 经过任意个可能的轨迹点, 且保持在道路 Lane 的 s 方向递增的最小 Cost 的轨迹曲线。从图 7-12 可以看出, 最小 Cost 轨迹所经过的轨迹点之间的连接是可以在进行图的遍历搜索的同时构建的。算法中 $g(n)$ 代表了到达节点 n 的 Cost, 而 $\phi(n)$ 代表了整个轨迹的 Cost, 其中包含了到达 n 的 Cost 及以 n 为轨迹终点带来的附加 Cost。前者 $g(n)$ 是选择从当前节点展开到后续节点所增加 Cost 的依据 (图 7-12 第 13 行); 而当选择从哪一个前驱节点到达当前节点时则考虑 $\phi(n)$ 作为评价标准 (图 7-12 第 11 行)。当整个 $g(n)$ 和 $\phi(n)$ 都计算完毕后, 很容易通过我们的前驱节点映射 `prev_node` 来倒推出整个最小 Cost 的轨迹点序列。我们只需要增加一个虚拟的节点 n_f , 且对于 n_f , 构建连接 n_f 的虚拟边。这样我们的任务便成为寻找一条连接 n_0 至 n_f 的轨迹点构成的最小 Cost 轨迹曲线。根据图 7-12 所示的算法, $g(n)$ 已经计算完毕, 那么最后一个实际的轨迹点可以从 $n_{\text{last}} = \underset{n}{\operatorname{argmin}} g(n) + \Phi(n)$ 中找出。

```

1 function Search_DP(TrajectoryPointMatrix(V,E), {s}, {l})
2   Initialize map g:  $\forall n \in V, g(n) \leftarrow \text{inf}$ 
3   Initialize map prev_node:  $\forall n \in V, g(n) \leftarrow \text{null}$ 
4   for each sampled  $s_i \in \{s\}$ :
5      $\forall n \in V$  s.t.  $s(n) = s_i: \phi(n) \leftarrow \text{inf}$ 
6     for each lateral direction Trajectory Point  $n = [s_i, l_j]$ :
7       if  $g(n) \neq \text{inf}$ :
8         Form the vehicle pose vector  $\hat{x}_n = [x(n), y(n), \theta(n), \kappa(n)]$ 
9         for each outgoing edge  $\tilde{e} = (n, n')$ 
10           Form the polynomial spiral  $\tau(\tilde{e}(n, n'))$ 
11           if  $g(n) + \Phi_c(n) < \phi(n')$ :
12              $\phi(n') \leftarrow g(n) + \Phi_c(n)$ 
13              $g(n') \leftarrow g(n) + c(\tau)$ 
14             prev_node( $n'$ )  $\leftarrow n$ 
15           end if
16         end for
17       end if
18     end for
19   end for

```

图 7-12 基于动态编程的轨迹点最小 Cost 轨迹算法

这里我们再简单介绍一下具体的轨迹曲线的 Cost 函数设计。考虑轨迹 $\text{Cost}(\Omega(\tau) = c(\tau) + \Phi(\tau))$ ，其中 $c(\tau)$ 部分的 Cost 代表连接曲线轨迹点的部分代价。这里可以考虑如下因素来设计基于轨迹的 Cost 函数。

(1) 道路层面：规划的轨迹曲线应当尽可能和道路的中心线切合。例如，当行为决策的输出是直行时，轨迹曲线的 Cost 会随着规划轨迹曲线偏离道路中心线的横向位移 l 加大而加大。

(2) 障碍物层面：规划的轨迹曲线需要避让静态障碍物。例如，在图 7-10SL 坐标系下道路的分割采样及可能的轨迹，需要将静态的障碍物所占领的网格及其附近的网格的 Cost 调整到非常高。我们将在 7.4.2 节中结合速度规划讨论动态障碍物的规避。

(3) 控制和舒适度层面：这个层面的 Cost 和操控的限制及乘客的舒适性紧密相关。规划的曲线应该尽量避免曲率（包括曲率导数）的较大变化等，使得乘客的舒适性得到保障。

再考虑整体曲线 Cost 的 $\Phi(\tau)$ 部分，由于我们将速度规划（Speed Planning）的部分作为一个单独的问题来解决，那么 $\Phi(\tau)$ 的 Cost 函数可以仅考虑轨迹曲线的纵向位移 s 部分。

$$\Phi(\tau) = -\alpha s_f(\tau) + h_d(s_f(\tau)),$$

$$h_d(s) = \begin{cases} -\beta & \text{if } s \geq s_{\text{threshold}} \\ 0 & \text{otherwise,} \end{cases}$$

其中第一项是线性 Cost，代表了整体轨迹曲线偏向于纵向位移 s 较长的轨迹（Cost 项为负代表对 Cost 进行减小 discount），而第二项是一个非线性 Cost，只有当整体的纵向位移 s 超过一定门限时才会触发。

7.4.2 速度规划

当轨迹规划给定了一条或者若干条选出的轨迹曲线后，动作规划模块需要解决的后续问题是在此轨迹的基础上加入速度相关的信息。这一问题我们称为速度规划。速度规划的目标是在给定的轨迹曲线上，在满足反馈控制的操作限制及符合行为决策的输出结果这两个前提下，将轨迹点赋予速度及加速度信息。我们已经将静态障碍物的规避部分在 7.4.1 节的轨迹规划中考虑，速度规划主要考虑的是对于动态障碍物的规避。本节中，我们引入 S-T 图这一概念，并且把无人车速度规划问题归纳成 S-T 图上的搜索问题进行求解。顾名

思义，S-T 图是一个关于时间和给定轨迹纵向位移的二维关系图。任何一个 S-T 图都是基于一条已经给定的轨迹曲线。根据无人车预测模块对动态障碍物的轨迹预测，每个动态障碍物都会在这条给定的轨迹上有所投影，从而产生对于一定 S-T 区域的覆盖。这里我们结合一个例子阐述基于 S-T 图的速度优化算法。

如图 7-13 所示，考虑一条轨迹规划选取的换道轨迹。此时，在需要换至的目标车道中有 a 和 b 两辆障碍车。简单而不失一般性，假设预测模块对这两辆车的预测轨迹都是沿着当前的左侧车道匀速直线行驶。那么 a 和 b 在这条选定的轨迹上的投影如图中的两块阴影区域所示。在某一个固定时刻， a 和 b 在轨迹上的投影都是平行于 s 轴的线段。随着 t 的延伸，障碍物 a 和 b 在轨迹曲线上的投影如图 7-13 所示的阴影四边形并不断延伸。我们将 S-T 图类似于轨迹规划的地图划分也切割成小网格 (Lattice Grid)，对每个网格赋予 Cost，那么速度规划问题也可以归纳成在这个网格图上的最小 Cost 路径搜索问题。主车在 $t=0$ 时刻在 $s=0$ 位置，主车需要最终到达 $s=s_{\text{end}}$ 位置且经过网格的累计 Cost 最低。

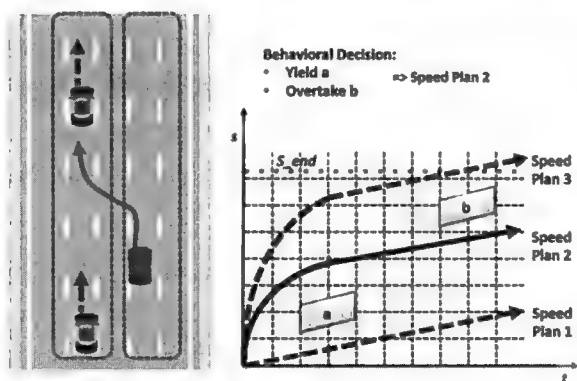


图 7-13 基于 S-T 图的速度规划搜索图

我们比较图 7-13 所示的三种速度规划方案：第一种方案 (Speed Plan 1) 在任意 t 时刻，主车在轨迹上的 s 方向一直落后于 a 和 b ，注意主车还有一个等于 Speed Plan 1 对应直线斜率的速度，最终可以到达 $s=s_{\text{end}}$ 位置 (图中的轨迹没有画出)，在实际行驶中，这个方案对应于让 a 和 b 都先通过主车换道需要经过的轨迹部分，然后再进行换道；第二种方案 (Speed Plan 2) 从某一时刻开始在轨迹上的 s 位移便一直领先于车辆 a ，但一直落后于车辆 b ，在实际行驶中对应先加速在 a 进入选定的换道轨迹前进入换道轨迹，但等待位置较为靠前的 b 先进入轨迹；第三种方案 (Speed Plan 3) 对应加速在 a 和 b 进入选定的换道轨迹前进入轨迹。此时，假设上游的行为决策模块的输出是针对障碍车辆 a 进行让

先 (Yield), 对障碍车辆 **b** 进行抢先 (Overtake), 那么速度规划算法应当结合 Cost 选出第二种方案 (Speed Plan 2)。

结合上游行为决策输出的信息, 动作规划模块的速度规划可以灵活设置障碍物体周边的 Cost, 达到调整速度方案的目的。例如, 当上游决定对于物体 **a** 进行抢先决策时, 在 S-T 图上物体 **a** 的运动轨迹上方的网格的 Cost 就可以调成偏小; 假设对一个动态障碍物体需要让先, 那么可以将该物体下方的网格的 Cost 调小。同时, 为了避免任何潜在的碰撞 (Collision), 所有动态障碍物体的轨迹经过的网格的 Cost 都需要调大。除此之外, 还需要考虑一条给定速度方案在加速度等控制方向的 Cost。例如, S-T 图上过“陡峭”的曲线代表加速度大甚至不连续, 这样很有可能导致反馈控制模块 (Feedback Control) 无法实际执行。所以, 每条曲线所代表的速度方案均有一个整体的 Cost。实际上, 如何根据上游输出和下游限制来调整 Cost, 是速度规划中的 S-T 图算法的关键设置。在设置好 Cost 的基础上, 最小 Cost 轨迹的产生可以用类似 A* 或者 Dijkstra 等简单搜索算法实现, 本节将不再赘述。在得到了最小 Cost 的 S-T 路径后, 可以简单算出任何一个轨迹位置的速度 (对应 S-T 图任意点斜率) 和加速度 (斜率的导数), 从而完成速度规划的计算。

结合 7.4.1 节中的轨迹规划和 7.4.2 节中的速度规划, 至此我们已经能够将无人车在基于周边环境和行驶目地下做出的行为层面决策, 通过一些列模块的计算转化成具体的带有位置、速度信息的时空轨迹点。我们将轨迹规划输出的曲线上按照均匀时间间隔提取抽样点, 并且将速度规划的结果信息填充到这些抽样点中。这些带有速度、加速度、角加速度的时空轨迹点将被发给下游的反馈控制模块, 进行无人车控制规划流程中的最后实际车辆执行的步骤。

7.5 反馈控制

单独从车辆的姿态控制的角度来看, 无人车反馈控制部分和普通的车辆反馈控制并无本质不同。二者都是基于一定的预设轨迹, 考虑当前车辆姿态和此预设轨迹的误差并进行不断的跟踪反馈控制。参考资料[12]中列出了很多关于无人车反馈控制的工作, 其中[10][12][14]等和传统的车辆反馈控制的不同之处在于在传统的反馈控制中加入了基于无人车对障碍物体的避让和路径的优化选择等。在本书提出的整个无人车规划控制的体系架构下, 我们的无人车反馈控制部分可以很大程度上借鉴传统的车辆姿态反馈控制的工作。由于这部分工作较为传统和成熟, 本书不做为重点来介绍, 我们在这里向读者介绍最重要

和基本的两个概念：基于车辆的自行车模型，以及 PID 反馈控制^{[15][16]}。其他车辆姿态反馈控制的工作读者可以参考其他文献。

7.5.1 自行车模型

为了更清楚地描述动作规划中的轨迹生成算法，我们对车辆的模型做了简单的介绍。这里我们更加详细地介绍一种无人车反馈控制（Feedback Control）中常用的车辆控制模型：自行车模型。自行车模型所代表的车辆姿态处于一个二维的平面坐标系内。车辆的姿态（pose）可以由车辆所处的位移（position）及车身和坐标平面的夹角（heading）完全描述。在该模型下，我们认为车辆前后轮由一个刚性（rigid）不变的轴连接，其中车辆的前轮可以在一定的角度范围内自由转动，而车辆的后轮保持和车身的平行关系不能转动。前轮的转动对应实际车辆控制中方向盘的转动。这种自行车模型的一个重要特征是：车辆无法在不做出向前移动的情况下进行横向位移。这种特征又称作非完整性约束（nonholonomic constraint）。在车辆模型中，这种约束根据坐标系的选择不同，往往以不同形式的车辆动作姿态微分方程的形式呈现。另外读者需要注意的是，为了使模型的计算简单，我们忽略车辆的惯性及轮胎接触地面点的打滑。在速度较低的情况下，惯性效应带来的误差较小可以忽略；但是在高速运动时，惯性效应对反馈控制的影响往往是不能忽略的。高速状态下考虑惯性的车辆动力学模型更复杂，不在本书讨论的范围之内。

车辆的自行车模型所代表的车辆姿态如图 7-14 所示。这里使用一个基于 x-y 的二维平面，其中 \hat{e}_x 和 \hat{e}_y 分别代表其 x 和 y 方向的单元向量。向量 p_r 和向量 p_f 分别代表车辆后轮和前轮与地面的接触点。车辆的朝向角 θ 代表车辆和 x 轴的夹角（即向量 p_r 和单元向量 \hat{e}_x 的夹角）。方向盘转角 δ 定义为前轮朝向和车辆朝向角的夹角。其中前后轮与地面接触点的向量 p_f 和 p_r 之间满足：

$$\begin{aligned}(\dot{p}_r \cdot \hat{e}_y) \cos(\theta) - (\dot{p}_r \cdot \hat{e}_x) &= 0 \\(\dot{p}_f \cdot \hat{e}_y) \cos(\theta + \delta) - (\dot{p}_f \cdot \hat{e}_x) \sin(\theta + \delta) &= 0\end{aligned}$$

其中 \dot{p}_f 和 \dot{p}_r 分别代表车辆前后轮在和地面接触点处的瞬时速度向量。考虑车辆的后轮速度在 x-y 轴的投影标量 $x_r := p_r \cdot \hat{e}_x$ 和 $y_r := p_r \cdot \hat{e}_y$ ，以及后轮的切向速度 $v_r := \dot{p}_r \cdot (p_f - p_r) / \|p_f - p_r\|$ ，那么上述的向量 p_f 和 p_r 之间的关系限制在后轮相关分量上的表现形式为：

$$\begin{aligned}\dot{x}_r &= v_r \cos(\theta) \\ \dot{y}_r &= v_r \sin(\theta) \\ \dot{\theta} &= v_r \tan(\delta) / l\end{aligned}$$

其中 l 代表车辆前后轴中心间距。类似地, 用车辆前轮相关分量的表现形式为:

$$\begin{aligned}\dot{x}_f &= v_f \cos(\theta + \delta) \\ \dot{y}_f &= v_f \sin(\theta + \delta) \\ \dot{\theta} &= v_f \sin(\delta) / l\end{aligned}$$

这里前后轮的切向速度标量大小满足: $v_r = v_f \cos(\delta)$ 。

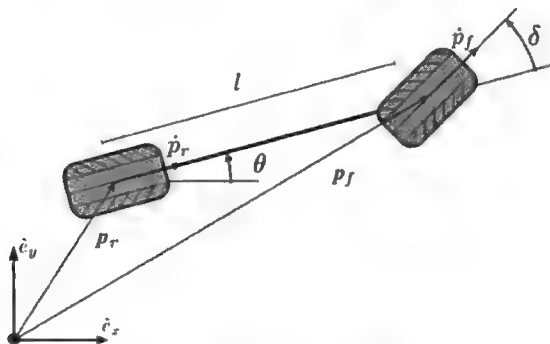


图 7-14 车辆控制的自行车模型^[4]

在上述车辆模型下, 反馈控制需要解决的问题之一便是找到满足车辆动态姿态限制的方向盘转角 $\delta \in [\delta_{\min}, \delta_{\max}]$ 及前向速度 $v_r \in [\delta_{\min}, \delta_{\max}]$ 。值得一提的是, 为了简化计算, 往往直接考虑朝向角的变化率 ω 而非实际的方向盘转角 δ 。这样便有 $\tan(\delta) / l = \omega / v_r = \kappa$, 问题简化为寻找满足条件的朝向角变化率, 而这样的近似常常被称为独轮车模型 (Unicycle Model), 其特点是前进速度 v_r 被简化为只与朝向角度变化率和轴长相关。

7.5.2 PID 反馈控制

一个典型的PID反馈控制系统的结果如图7-15所示。其中 $e(t)$ 代表当前的跟踪误差, 而这个跟踪的变量误差可以是轨迹的纵向/横向误差、角度/曲率误差或者是若干车辆姿态状态变量的综合误差。其中 P 控制器代表对当前误差的反馈, 其增益由 K_P 控制; I 和 D 控制器分别代表积分项和微分项, 其增益分别由 K_I 和 K_D 控制。

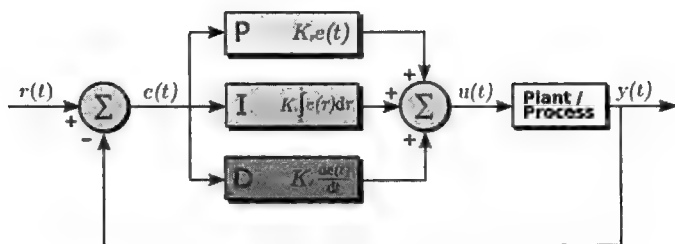


图 7-15 基于 PID 的反馈控制系统^[15]

具体到无人车的反馈控制模块,我们需要解决的问题是控制车辆尽可能遵循上游动作规划所输出的时空轨迹。这里我们借鉴参考资料[12]中的思路,使用两个基于 PID 反馈控制的控制器分别控制方向盘转角 δ 及前进速度 v_s 。其中在 n 采样时刻,控制方向盘转角的 PID 控制器如下:

$$\delta_n = K_1 \theta_e + K_2 l_e / V_s + K_3 \dot{l}_e + K_4 \sum_{i=1}^n l_e \Delta t$$

其中 θ_e 代表当前车辆朝向和动作规划输出的基准轨迹点 (Reference Point) 之间的跟踪角度误差, l_e 代表在横向位置相对于基准轨迹点 (Reference Point) 的误差, V_s 代表车辆在纵向方向的速度。在车辆纵向方向的 PID 控制器主要考虑车辆的轨迹曲率 k_{Vehicle} 和动作规划输出的基准点曲率 $k_{\text{Reference}}$ 。根据曲率,我们可以设计一个跟踪速度误差的函数 $f(k_{\text{Vehicle}}, k_{\text{Reference}})$, 那么在纵向的目标速度变为: $V_{\text{desired}} = V_s - f(k_{\text{Vehicle}}, k_{\text{Reference}})$ 。根据此目标速度和当前车辆姿态的前进速度 V_s , 前进速度的 PID 控制器可以写成:

$$\begin{aligned} V_e &= V_{\text{desired}} - V_s \\ U_V &= K_P V_e + K_I \sum V_e \Delta t + K_D \Delta V_e / \Delta t \end{aligned}$$

其中 K_P 、 K_I 和 K_D 分别代表当前比例项、积分项和微分项的增益, U_V 代表该采样周期输出的油门控制反馈。

以上两种对于方向盘和油门分别设计的 PID 控制器可以认为是最基本的无人车反馈控制的实现单元。为了使行驶过程更加平滑顺畅舒适,往往在曲率误差较大时,需要设计更复杂的反馈控制系统对车辆进行控制。对于车辆按照预设轨迹的精确控制并不是无人驾驶的特有问题,且已经有很多可行的解决方案,感兴趣的读者可以参考参考资料[12][17]。

7.6 无人车规划控制结语

作者认为整个无人车广义规划控制范畴下的路由寻径、行为决策、动作规划及反馈控制等几大模块,在当前的学术界和工业界都有一些较成熟的解决方案可以借鉴。这些解决方案有些有着牢固的理论基础和数学推导,还有的在实际的无人车相关工程实践中有着出色的表现。事实上,我们认为单独看控制规划每个层面需要解决的问题,都不是非常困难。如何将整个无人车规划控制的问题有效清晰地划分到不同的模块,并且将各个上下游模块的解决方案配合起来达到整体的协调效果,才是无人车广义规划控制的难点和挑战所在。从这个角度讲,本书并不着力于以调研的形式介绍所有模块层面的现存解决方案,而是着眼于清晰地提供一套有效划分无人车控制规划这一复杂问题到不同层面子问题的方法。我们试图向读者展示,如何有效地将无人车控制规划这样一个复杂问题自上而下地进行分割,并且明确每个层面需要解决的具体问题的范围和限制。在这个基础上,我们再详细介绍每个模块的一种或几种可行的解决算法。我们希望通过展示这样的“分而治之”的控制规划解决思路,有益于读者对整个无人车控制规划系统运作的了解。

7.7 参考资料

- [1] Proactive MDP-based Collision Avoidance Algorithm for Autonomous Cars. D, Osipych, D. Tran, W. Sheng, G. Chowdhary, R. Zeng. *IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems (CYBER)*, 2015.
- [2] Probabilistic MDP-Behavior Planning for Cars. S. Brechtel, R. Dillmann. *IEEE Conference on Intelligent Transportation Systems*. October 2011.
- [3] Probabilistic Online POMDP Decision Making for Lane Changes in Fully Automated Driving. S. Ulbrich and M. Maurer. 16th International Conference on Intelligent Transportation Systems (*ITSC*), 2013.
- [4] A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles. B. Paden, M. Cap, S.Z. Yong, D. Yershow and E. Frazzolo. *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33-55, 2016.
- [5] The DARPA Urban Challenge: Autonomous Vehicles in City Traffic. Editors:

- Buehler, Martin, Iagnemma, Karl, Singh, Sanjiv(Eds.) 2009.
- [6] Junior: The Stanford Entry in the Urban Challenge. M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhneke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, S. Thrun. *Journal of Field Robotics: Special Issue on the 2007 DARPA Urban Challenge*. Volume 25, Issue 9, September 2008, Pages 569-597.
 - [7] https://en.wikipedia.org/wiki/PID_controller.
 - [8] Dijkstra's Algorithm. https://en.wikipedia.org/wiki/Dijkstra's_algorithm.
 - [9] A* Algorithm. <http://web.mit.edu/eranki/www/tutorials/search/>.
 - [10] Focused Trajectory Planning for Autonomous On-Road Driving. T. Gu, J. Snider, J. M. Dolan, J. Lee. *IEEE Intelligent Vehicles Symposium (IV)*, 2013.
 - [11] Parallel Algorithms for Real-time Motion Planning. Matthew McNaughton. Doctoral Dissertation. *Robotics Institute, Carnegie Mellon University*, 2011.
 - [12] Dynamic Curvature Steering Control for Autonomous Vehicle: Performance Analysis. M.A. Zakaria, H. Zamzuri, S.A. Mazlan. *IOP Conference series: Materials Science and Engineering* 114(2016) 012149.
 - [13] Polynomial Spiral. <http://web.calstatela.edu/curvebank/waldman4/waldman4.htm>.
 - [14] Trajectory Generation and Control Methodology for an Ground Autonomous Vehicle. J. Connors and G.H. Elkaim. *AIAA Guidance, Navigation and Control Conference*.
 - [15] Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. Christos Katrakazas, Mohammed Quddus, Wen-Hua Chen, Lipika Deka. *Elsevier Transportation Research Part C: Emerging Technologies*. Volume 60, November 2015, Pages 416-442.
 - [16] National Instruments: <http://www.ni.com/white-paper/3782/en/>.
 - [17] 基于四阶贝塞尔曲线的无人车可行轨迹规划. 陈成, 何玉庆, 卜春光, 韩建达. 自动化学报, 2015, 41(3): 486-496.

- [18] Autonomous Driving in Urban Environments: Boss and the Urban Challenge, C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M.N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T.M. Howard, S.Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y.W. S, S. Singh, J. Snider, A. Stentz, W. Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, D. Ferguson. *Journal of Field Robotics: Special Issue on the 2007 DARPA Urban Challenge*. Volume 25, Issue 9, September 2008, Pages 425-466.
- [19] Earth Coordination System https://en.wikipedia.org/wiki/Geographic_coordinate_system.

8

基于 ROS 的无人驾驶系统

本章着重介绍基于机器人操作系统 ROS 的无人驾驶系统。无人驾驶系统是个十分复杂的软硬件系统,为了支持这个复杂的系统更有效的运行,我们需要一个操作系统去整合、管理,以及调度不同的模块。ROS 作为一个相对成熟的机器人操作系统,很适合使用在无人驾驶系统中。本章作者将介绍机器人操作系统 ROS 及 ROS 在无人驾驶场景上的优缺点,以及讨论如何在 ROS 的基础上提升无人驾驶系统的可靠性、通信性能及安全性。

8.1 无人驾驶:多种技术的集成

无人驾驶技术是多个技术的集成,如图 8-1 所示,一个无人驾驶系统包含了多个传感器,包括长距雷达、激光雷达、短距雷达、摄像头、超声波、GPS、陀螺仪等。每个传感器在运行时都不断产生数据,而且系统对每个传感器产生的数据都有很强的实时处理要求。例如摄像头需要达到 60 FPS 的帧率,意味着留给每帧的处理时间只有 16 毫秒,但当数据量增大之后,分配系统资源便成了一个难题。例如,当大量的激光雷达点云数据进入系统,占满 CPU 资源,就很可能令摄像头的的数据不能得到及时处理,导致无人驾驶系统错过交通灯的认识,造成严重后果。

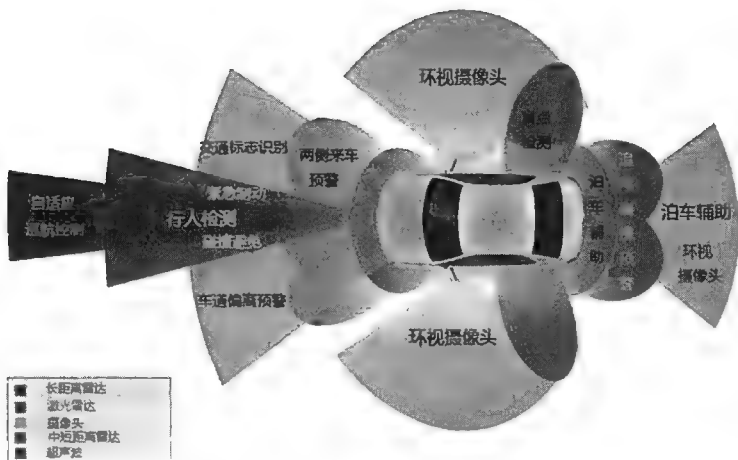


图 8-1 基于多传感器融合的无人驾驶系统范例

如图 8-2 所示，无人驾驶系统整合了多个软件模块，包括有路径规划、避障、导航、交通信号监测等，以及多个硬件模块，包括计算模块、控制模块、传感器模块等，如何有效地调配软硬件资源是一个挑战。具体有如下几个问题：第一，软硬件模块数据增加，运行期间难免有些模块会出现异常退出，或者导致系统崩溃。在这种情况下，如何为系统提供自修复能力？第二，由于模块之间有很强的联系，如何管理模块间的有效通信？对于关键模块间的通信，信息不可丢失，不可有过大的延时。第三，每个功能模块间如何进行资源隔离，如何分配计算与内存资源，当资源不足时如何确认更高的优先级执行？

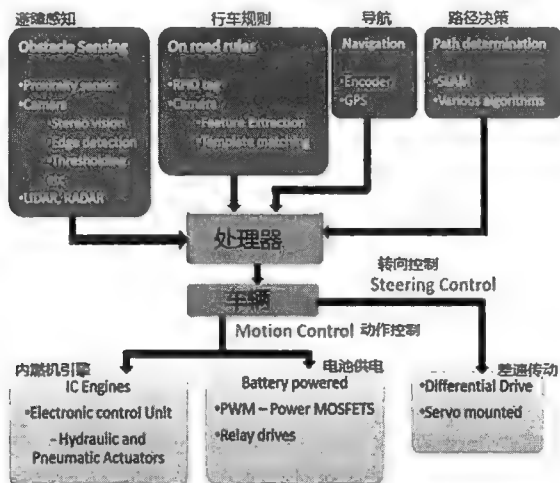


图 8-2 无人驾驶软硬件整合结构图

简单的嵌入式系统并不能满足上述无人驾驶系统的需求，我们需要一个成熟、稳定、高性能的操作系统来管理各个模块。在经过详细调研后，我们觉得机器人操作系统 ROS 比较适合无人驾驶场景。在 8.2 节中我们会介绍 ROS 的优缺点，以及在 ROS 上增加什么功能使之更适用于无人驾驶系统。

8.2 机器人操作系统（ROS）简介

ROS 是 Willow Garage 公司 2010 年发布的开源机器人操作系统，由于其具有点对点设计、不依赖编程语言、开源等特点，短短几年时间 ROS 便成为了全世界机器人研究的热门仿真开发操作平台。^[1] ROS 之所以被称为操作系统，是因其具有与操作系统类似的硬件抽象、底层驱动管理、消息传递等功能，然而它并不是真正意义上的操作系统，只能算是中间件。ROS 具有很强的代码可复用性和硬件抽象性能，采用分布式架构，通过各功能独立的节点（Node）实现消息传递任务的分层次运行，从而减轻实时计算的压力。此外，ROS 是一个强大并且灵活的机器人编程框架，ROS 为常用的机器人和传感器提供了硬件驱动接口。从软件架构角度讲，它是一种基于消息传递通信的、分布式多进程框架。ROS 很早就被机器人行业使用，很多知名的机器人开源库，例如基于 quaternion 的坐标转换、3D 点云处理驱动、规划方面的 MoveIt、OpenRAVE 规划库、控制方面的 OROCOS 实时运动控制库、视觉图像处理方面的 OpenCV 和 PCL 开源库、定位算法 SLAM 等都是开源贡献者基于 ROS 开发的。

8.2.1 ROS 中的基本组成

ROS 中最重要的概念包括节点、节点管理器、参数服务器、消息、主题、服务和任务等。

（1）节点（Node）：节点是用来实现运算功能的进程，ROS 机器人仿真框架由功能独立的节点组成。移动操作机械臂仿真时，激光距离传感器节点用来读取激光数据，电机控制节点用来读取电机信息并控制电机转动，路径规划节点用来实现移动平台的运动轨迹规划，特定功能的节点各司其职，从而构成了完整的机器人仿真系统。

（2）节点管理器（Master）：顾名思义，节点管理器主要用来管理节点。每个节点都需要通过节点管理器实现节点名字的注册，节点之间的相互查找也需要在节点管理器内进行。缺少了节点管理器，节点间将不能进行信息传输，服务和任务都将无法找到服务器。

(3) 参数服务器 (Parameter Server): 参数服务器的主要目的是节点运行时用来存取参数, 它并不是用来实现高效能的数据传输, 而是用来存放节点运行所需的配置参数。

(4) 消息 (Message): 节点之间的通信内容称为消息。一个消息是一个由类型域构成的简单的数据结构。它以基本型的阵列形式支持标准的原始属于类型 (像整型、浮点型、布尔型等)。

(5) 主题 (Topic): 节点之间的消息传输围绕一个特定的主题。主题名称就是传输消息的主要内容。如图 8-3 所示, 在特定的主题下, 节点可以发布满足消息类型要求的消息, 当其他节点需要该话题的消息内容时, 它只需要创建接收器并接收该主题即可。一个节点可以同时发布和接收多个主题。主题发布者和主题接收者不知道对方的存在, 通过节点管理器, 节点发布者能够获知当前主题接收节点的个数。主题的发布/接收模式是一种弹性的异步通信方式, 类似于围绕共同主题的多人聊天室, 每个人都可以自由地发表符合主题要求的消息, 然而这种方式并不能保证发布的消息能够得到及时响应。

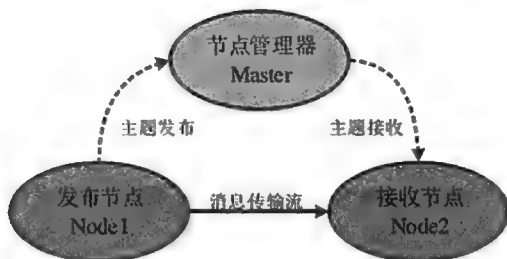


图 8-3 ROS 主题发布/接收机制示意图

1. 服务 (Service)

如图 8-4 所示, ROS 中的服务是一种利用同步通信的方式请求/回复交互的分布式系统。提供服务的节点称为服务器端, 其他节点通过发送满足服务请求格式要求的消息使用服务, 它们被称为客户端。因此, 服务需要请求和回复这一对消息结构。

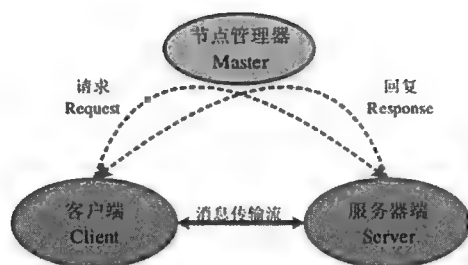


图 8-4 ROS 服务请求/回复机制示意图

2. 任务 (Action)

ROS 中的 Action 功能包主要用来实现服务器端 (Server) 和客户端 (Client) 之间的信息交换。如图 8-5 所示，任务机制中包含了五个基本的主题，箭头的方向指示了消息的传输方和接收方。客户端向服务器端发布机器人期望状态信息，轨迹跟踪控制服务器端接收到目标指令后，将发布命令启动轨迹跟踪控制器并能够实时获得机器人的当前状态信息，服务器端会向客户端发送实时状态反馈及最终执行结果。在当前任务完成以后，便可以立即完成下一个任务。如果需要取消当前任务，客户端可以向服务器端发送取消指令并重新发送新的目标，如果不取消当前任务直接发送新任务，服务器端会按照优先级顺序完成任务。

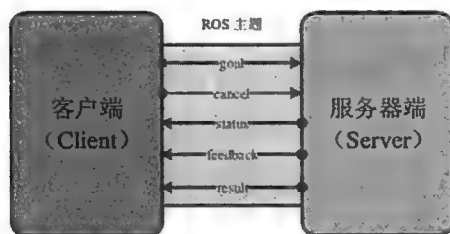


图 8-5 客户端与服务器端的消息传输方式示意图

因为 ROS 本身是基于消息机制的，这样的做法使得模块开发者可以根据软件的功能把软件拆分成各个模块，每个模块只负责读取消息和分发消息，每个模块通过消息关联。如图 8-6 所示，最左边的节点可能会负责从硬件驱动读取数据 (比如 Kinect)，读出的数据会以消息的方式打包，ROS 底层会识别这个消息的使用者，然后把消息数据分发给他们。

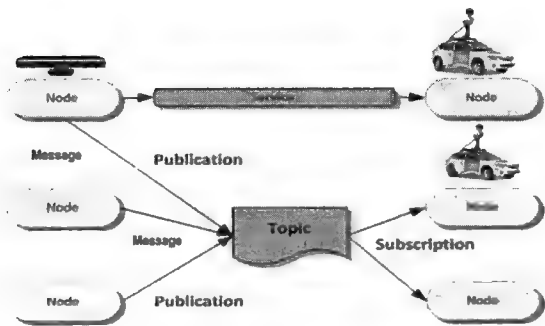


图 8-6 ROS 系统结构原理

8.2.2 ROS 1.0 Vs. ROS 2.0

ROS 1.0 起源于 Willow Garage 的 PR2 项目，主要的部件分为 3 种：ROS Master、ROS Node 和 ROS Service。ROS Master 的主要功能是命名服务，它存储了启动时需要的运行时参数，消息发布上游节点和接收下游节点的连接名和连接方式，和已有 ROS 服务的连接名。ROS Node 是真正的执行模块，对于收到的消息进行处理，并且发布新的消息给下游节点。ROS Service 是一种特殊的 ROS 节点，相当于一个服务节点，接受请求并返回请求的结果。图 8-7 展示了 ROS 通信的流程顺序，首先节点会向 master 节点发布（advertise）或者订阅（subscribe）感兴趣的主体（topic）。当创建连接时，下游节点会向上游节点 TCP Server 发布连接请求，等连接创建后，上游节点的消息会通过连接送至下游节点。

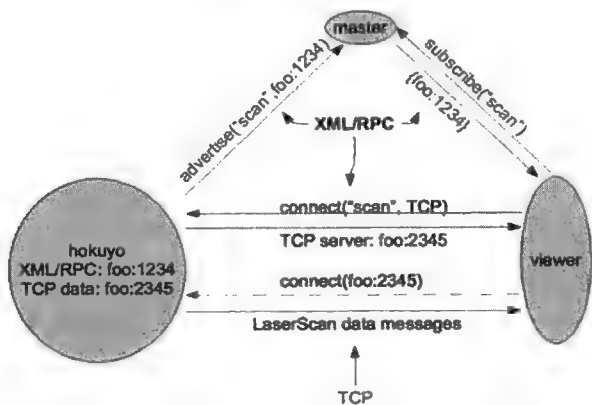


图 8-7 ROS Master Node 通信模型

ROS 2.0 主要是为了让 ROS 能够符合工业级的运行标准,这里主要采用了 DDS 这个工业级别的中间件负责可靠通信,通信节点动态发现,而且用 shared memory 方式使得通信效率更高。使用 DDS 以后,所有节点的通信拓扑结构都是依赖于动态 P2P 的自发现模式,也就去掉了 ROS Master 这个中心节点。如图 8-8 所示,RTI Context、PrismTech OpenSplice 和 Twin Oaks 都是 DDS 的中间件提供商,上层通过 DDS API 封装,这样 DDS 的实现对于 ROS Client 透明。在设计上,ROS 主页详细讨论了用 DDS 的原因 http://design.ros2.org/articles/ros_on_dds.html。

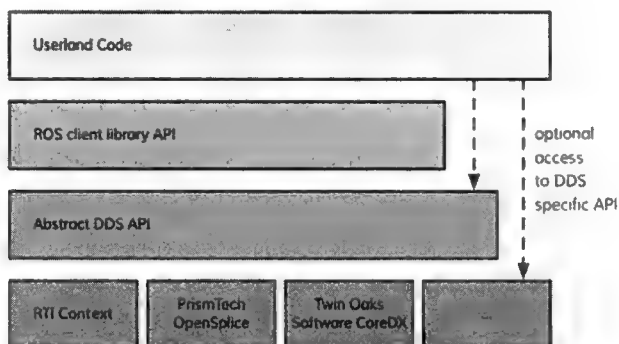


图 8-8 ROS 2.0 DDS 系统分层示意图

在无人车驾驶系统中,我们选择 ROS1.0 作为我们的开发平台,之所以选择 ROS 1.0 而不是 ROS 2.0 主要有以下几点考虑。

(1) ROS 2.0 还是一个开发中的框架,很多功能不是很完整,需要更多测试与验证。在无人驾驶环境中,稳定性与安全性是至关重要的,我们需要基于一个已经经过验证的稳定系统来保证系统的稳定性和安全性,并提升其性能以达到无人车的要求。

(2) DDS 本身的耗费。我们测试了在 ROS 1.0 上直接使用 DDS 中间件的性能代价。国防科技大学有一个开源项目 MicROS (<https://github.com/cyberdb/micROS-drt>) 在这方面已经做了相关的尝试。实验发现,在一般的 ROS 通信场景中(100K 发送者接收者通信),ROS on DDS 的吞吐率并不及 ROS1.0。主要原因是 DDS 本身框架的耗费比 ROS 多一些,同时用了 DDS 以后的 CPU 占用率有明显提高。但是我们也确认了使用 DDS 之后,ROS 的 QoS 高优先级的吞吐率和组播能力有了大幅提升。我们的测试基于 PrismTech OpenSplice 的社区版,在它的企业版中有针对单机的优化,比如使用了共享内存的优化,这个我们暂未具体测量。

DDS 接口的复杂性。DDS 本身就是一套庞大的系统，其接口定义极其复杂，同时，文档支持较薄弱，这也是我们不想直接使用的一个原因。

8.3 系统可靠性

如上文所述，系统可靠性是无人驾驶系统最重要的特性。试想几个场景：第一，系统运行时 ROS 的 MASTER 出错退出，导致系统崩溃；第二，其中一个 ROS 的节点出错，导致系统部分功能缺失。以上任何一个场景在无人驾驶环境中都可能造成严重的后果。对 ROS 而言，其在工业领域的应用可靠性是非常重要的设计考量，但是目前的 ROS 设计对这块考虑得比较少，本节将讨论实时系统的可靠性所涉及的一些方面。

8.3.1 去中心化

ROS 的重要节点需要热备份，宕机可以随时切换。在 ROS 1.0 的设计中，主节点维护了系统运行所需的连接信息、参数信息及主题信息，如果 ROS Master 宕机了，整个系统就有可能无法正常运行。去中心化的解决方案有很多，如图 8-9 所示，为了解决这个问题，我们可以采用类似 ZooKeeper 采用主从节点的方式，同时主节点的写入信息随时备份，主节点宕机后，备份节点被切换为主节点，并且用备份的主节点完成信息初始化。^[2]

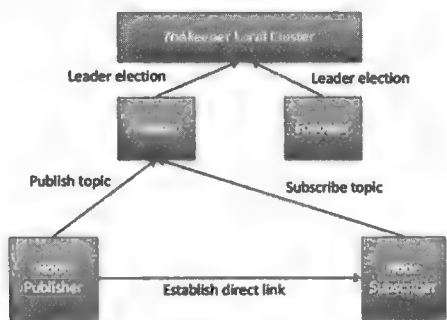


图 8-9 基于 ZooKeeper 的监控和报警原理示意图

8.3.2 实时监控和报警

对于运行的节点实时监控其运行数据，并在必要时报警。目前，ROS 并没有针对监控做太多的构架考虑，然而这块恰恰是最重要的。对于运行时的节点，监控其运行数据比

如应用层统计信息、运行状态等都对将来的调试、错误追踪有很多好处。在检测到严重的错误信息时必须报警。如图 8-10 所示，从软件构架层上来说主要分成 3 部分：ROS 节点层的监控数据 API，让开发者能够设置所需的统计信息，通过统一的 API 进行记录；监控服务端定期从节点获取监控数据（对于紧急的报警信息，节点可以把消息推送给监控服务端）；获取到监控数据后，监控服务端对数据进行整合、分析和记录，在察觉到异常信息后就会报警。

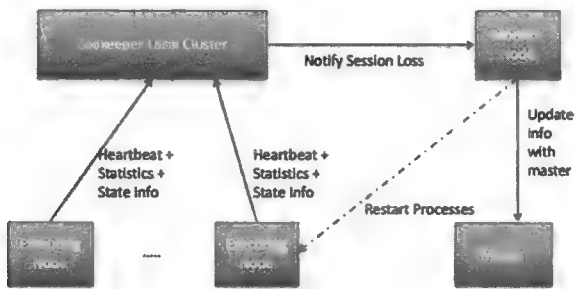


图 8-10 基于 ZooKeeper 的监控和报警原理示意图

8.3.3 节点宕机状态恢复

节点宕机时，需要通过重启的机制恢复节点，这个重启可以是无状态的，但有时也必须是有状态的，因此状态的备份格外重要。节点的宕机检测也是非常重要的，如果察觉到节点宕机，必须很快使用备份的数据重启。这个功能我们已经在 ZooKeeper 框架下实现了。

8.4 系统通信性能提升

由于无人驾驶系统模块很多，模块间的信息交互很频繁，提升系统通信性能会对整个系统性能有很大提升。我们主要从三个方面提高性能。

第一，目前同一个机器上的 ROS 节点间的通信是使用网络栈的 loop-back 机制，也就是说每一个数据包都需要经过多层软件栈处理，这将造成不必要的延时（每次 20 微秒左右）与资源消耗。为了解决这个问题，我们可以使用共享内存的方法把数据 memory-map 到内存中，然后只传递数据的地址与大小信息，从而把数据传输延时控制在 20 微秒内，并且节省了许多 CPU 资源。

第二，现在 ROS 做数据 broadcast 的时候，底层实现其实是使用 multiple unicast，也就是多个点对点的发送。假如要把数据传给 5 个节点，那么同样的数据会被复制 5 份。这造成了很大的资源浪费，特别是内存资源的浪费。另外，这样也会对通信系统的吞吐量造成很大压力。为了解决这个问题，我们使用了组播 multicast 机制：在发送节点和每一接收节点之间实现点对多点的网络连接。如果一个发送节点同时给多个接收节点传输相同的数据，也只需复制一份相同的数据包即可。它提高了数据传送效率，减少了骨干网络出现拥塞的可能性。图 8-11 对比了原有的通信机制与组播机制的性能，随着接收节点数量的增加（X 轴），原有的通信机制的数据吞吐量急剧下降，而组播机制的数据吞吐量比较平稳，没有受到严重影响。

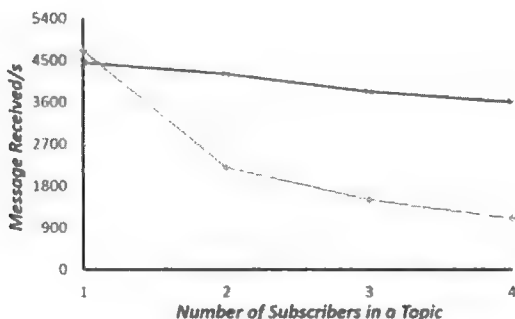


图 8-11 Multicast 性能提升

第三，对 ROS 的通信栈研究后我们发现，通信的延时很大的损耗是在数据的序列化与反序列化的过程中。序列化是将内存里对象的状态信息转换为可以存储或传输的形式。在序列化期间，对象将其当前状态写入临时或持久性存储区。以后，可以通过从存储区中读取或反序列化对象的状态重新创建该对象。为了解决这个问题，我们使用了轻量级的序列化程序，将序列化的延时降低了 50%。

8.5 系统资源管理与安全性

我们现在可以想象两个简单的攻击场景。第一，其中一个 ROS 的节点被劫持，然后不断地进行内存分配，导致其系统内存消耗殆尽，造成系统 OOM 开始关闭不同的 ROS 节点进程，使整个无人驾驶系统崩溃。第二，ROS 的 topic 或者 service 被劫持，导致 ROS 节点之间传递的信息被伪造，从而导致无人驾驶系统的异常行为。

如何解决资源分配与安全问题是无人驾驶技术的一个大课题。我们选择的方法是使用 Linux Container (LXC) 管理每一个 ROS 节点进程。^[1] 简单来说, LXC 提供轻量级的虚拟化, 以便隔离进程和资源, 而且不需要提供指令解释机制及全虚拟化等其他复杂功能, 相当于 C++ 中的 NameSpace。LXC 有效地将由单个操作系统管理的资源划分到孤立的群组中, 以更好地在孤立的群组之间平衡有冲突的资源使用需求。对于无人驾驶场景来说, LXC 最大的好处是性能损耗小。我们测试发现, 在运行时, LXC 只造成了 5% 左右的 CPU 损耗。

除了资源限制外, LXC 也提供了沙盒支持, 使得系统可以限制 ROS 节点进程的权限。为了避免可能有危险性的 ROS 节点进程破坏其他的 ROS 节点进程运行, 沙盒技术可以限制可能有危险性的 ROS 节点进程访问磁盘、内存及网络资源。另外, 为了防止节点中的通信被劫持, 我们还实现了节点中通信的轻量级加密解密机制, 使得黑客不可以回放或更改通信内容。

8.6 结论

一个复杂的系统需要一个成熟有效的管理机制保证其运行的稳定与高效, 使得系统中每个模块发挥出最大的潜能。在无人驾驶场景中, ROS 提供了这样一个管理机制, 使得系统中的每个软硬件模块都能有效的互动。原生的 ROS 提供了许多必要的功能, 但是这些功能并不能满足无人驾驶的所有需求, 因此我们在 ROS 之上进一步提高了系统的性能与可靠性, 完成了有效的资源管理及隔离。随着无人驾驶技术的发展, 相信更多的系统需求会被提出, 比如车车互联、车与城市交通系统互联、云车互联、异构计算硬件加速等, 我们也将持续优化我们的系统, 力求让此系统变成无人驾驶的标准系统。

8.7 参考资料

- [1] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Ng, "ROS: An open-source robot operating system", *Proc. Open-Source Software Workshop Int. Conf. Robotics and Automation*, 2009.
- [2] P. Hunt, M. KONAR, F. P.JUNQUEIRA, AND B. REED, ZooKeeper: Wait-free Coordination for Internet-Scale Systems. *In Proc. Usenix Annual Technical*

Conference, June 2010.

- [3] K.-T. Seo, H.-S.Hwang, I.-Y.Moon, O.-Y.Kwon, B.-J.Kim, Performance comparison analysis of linux container and virtual machine for building cloud, *Advanced Science and Technology Letters*, vol. 66, pp. 105-111, 2014.

无人驾驶的硬件平台

本章将着重介绍无人驾驶的硬件平台设计。无人驾驶系统是多种技术、多个模块的集成，其中包括传感器平台、计算平台和控制平台。本章先介绍传感器平台，这是无人驾驶系统智能的关键所在。然后介绍激光雷达、毫米波雷达、车载摄像头、GPS、陀螺仪、V2X 等现有传感器解决方案，从技术原理、产品分类、行业现状等多方面讨论传感器平台在无人驾驶中的应用与发展。计算平台是这个复杂系统的大脑，目前正值无人驾驶的高速发展期，业界在面向无人驾驶计算的专有芯片设计及选择上也是百花齐放，有基于 CPU、GPU、FPGA、DSP，以及 ASIC 驾驶器等多种解决方案。本章将分析无人驾驶任务的计算需求，以及每种芯片的优缺点，然后基于分析讨论适合无人驾驶的芯片设计方案。控制平台是无人车的核心部件，主要包括电子控制单元与通信总线两大部分，其中 ECU 主要实现控制算法，通过对采集的各机械部件传感器信号进行运算比较，完成对控制部件多项参数的控制与设置；通信总线则主要实现 ECU 与机械部件间的通信，通过不同协议的通信总线有效地解决线路信息传递中的复杂化问题，从而实现对整车多种控制系统的总控。

9.1 无人驾驶：复杂系统

无人驾驶技术是多个技术的集成，如图 9-1 所示，一个无人驾驶系统包含了多个传感器，包括长距雷达、激光雷达、短距雷达、车载摄像头、超声波、GPS、陀螺仪等。每个传感器在运行时都不断产生数据，而且系统对每个传感器产生的数据都有很强的实时处理

要求。例如，摄像头需要达到 60 FPS 的帧率，意味着留给每帧的处理时间只有 16 毫秒，当数据量增大之后，分配系统资源便成了一个难题。例如，当大量的激光雷达点云数据进入系统，占满 CPU 资源，很可能令摄像头的的数据不能得到及时处理，导致无人驾驶系统错过交通灯的识别，造成严重后果。因此，合理地选择计算平台完成实时的大规模传感数据处理，进行实时的驾驶预警与决策，对无人驾驶的安全性、可靠性、持续性至关重要。在提供高性能的数据处理支持的同时，计算平台还需要兼顾功耗、散热、硬件体积等问题，这对于持续的安全行驶同样重要。因此，在现有无人车计算平台中，各种硬件模块都有相关集成解决方案。不同的计算单元通过 Switch 或者 PCIe Switch 相连，进行数据交换，完成协同运算。无人驾驶中除了需要对智能驾驶相关的传感器数据进行计算与决策，还需要传统汽车中各个机械部件进行配合控制，完成驾驶操作的执行与转换。这就需要控制平台：ECU 与通信总线的协助。ECU 从用途上讲是汽车专用微机控制器，它使用一套以精确计算和大量实验数据为基础的固定程序，不断地比较和计算各个机械部件传感器的数据，然后发出指令，完成机械控制。通信总线如 CAN、USB3.0、LIN 等则是在这个过程中实现汽车数据共享及指令的有效传达。

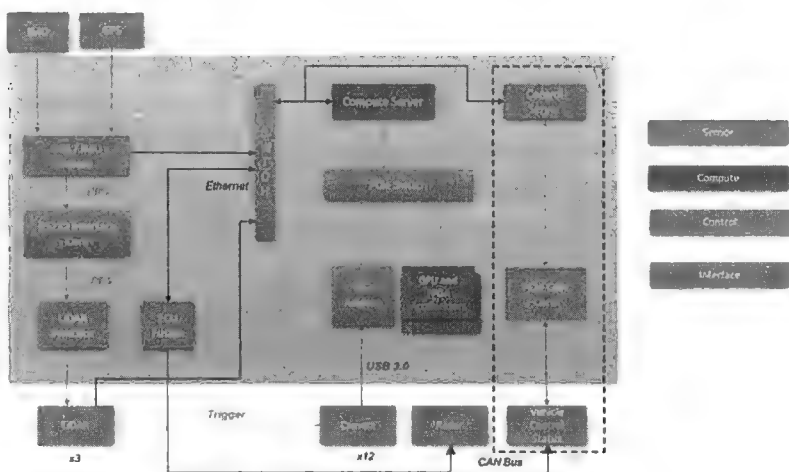


图 9-1 无人驾驶硬件平台示例图

9.2 传感器平台

目前，现有的车载传感器包括超声波雷达、激光雷达、毫米波雷达、车载摄像头、红外探头等。主流的无人驾驶传感平台以雷达和车载摄像头为主，并呈现多传感器融合发展

的趋势。基于测量能力和环境适应性，预计雷达和车载摄像头会持续传感器平台霸主的地位，并不断地与多种传感器融合，发展出多种组合版本。

表 9-1 中我们给出了现有的多种传感器在远距离测量能力、分辨率、温度适应性等诸多无人驾驶关键特性上的性能表现，可见各个传感器各有优劣，无法在单传感器的情况下完成对无人驾驶功能性与安全性的全面覆盖，这也显示了多传感器融合的必要性。因此，完备的无人驾驶系统应该如图 9-2 所示，各个传感器之间借助各自所长相互融合、功能互补、互为备份、互为辅助。

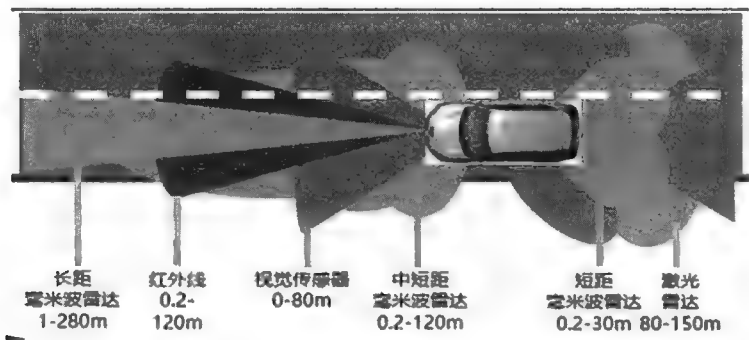


图 9-2 各种传感器在无人驾驶中的应用

表 9-1 各种车载传感器的性能对比

	激光雷达	毫米波雷达	摄像头	GPS/IMU
远距离测量能力	优	优	优	优
分辨率	良	优	优	优
低误报率	良	优	一般	优
温度适应性	优	优	优	优
不良天气适应性	较差	优	较差	优
灰尘/潮湿适应性	较差	优	较差	较差
低成本硬件	较差	优	优	良
低成本信号处理	较差	优	较差	良

9.2.1 激光雷达

激光雷达的工作原理是利用可见和近红外光波（多为 950nm 波段附近的红外光）发射、反射和接收来探测物体。激光雷达可以探测白天或黑夜下的特定物体与车之间的距离。

由于反射度的不同,也可以区分车道线和路面,但是无法探测被遮挡的物体和光束无法达到的物体,在雨雪雾天气下性能较差。

激光雷达在无人驾驶运用中拥有两个核心作用。

(1) 3D 建模进行环境感知。通过雷射扫描可以得到汽车周围环境的 3D 模型,运用相关算法比对上一帧和下一帧环境的变化可以较容易地探测出周围的车辆和行人。

(2) SLAM 加强定位。3D 雷射雷达的另一大特性是同步建图 (SLAM),实时得到的全局地图通过和高精度地图中特征物的比对,可以实现导航及加强车辆的定位精度。

1. 激光雷达的分类与产品

LIDAR 以单线/多线及距离两大因素为标准,价格从几百美元到几万美元不等。单线激光雷达的应用在国内已相对较广,像扫地机器人使用的便是单线激光雷达。单线激光雷达可以获取 2D 数据,但无法识别目标的高度信息,而多线激光雷达则可以识别 2.5D 甚至是 3D 数据,在精度上会比单线雷达高很多。目前,在国际市场上推出的主要有 4 线、8 线、16 线、32 线和 64 线。随着线速的提升,其识别的数据点也随之增加,所要处理的数据量也非常巨大。例如,Velodyne 的 HDL-32E 传感器每秒能扫描 70 万个数据点,而百度无人车和谷歌无人车配备的 Velodyne HDL-64E 通过 64 束的雷射束进行垂直范围 26.8 度、水平 360 度的扫描,每秒能产生的数据点高达 130 万。Velodyne HDL-64E 的内部结构如图 9-3 所示,主要由上下两部分组成。每部分都发射 32 束的雷射束,由两块 16 束的雷射发射器组成,背部包括信号处理器和稳定装置。

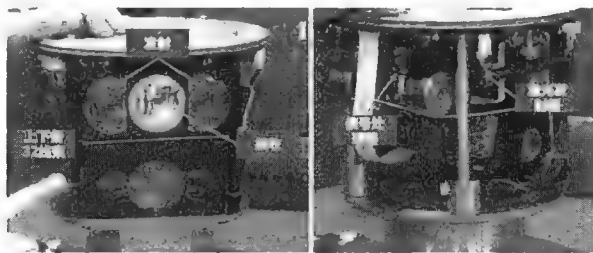


图 9-3 激光雷达结构示意图

激光雷达雷射发射器线束越多,每秒采集的云点就越多。线束越多就代表雷射雷达的造价越昂贵,以 Velodyne 的产品为例,64 线束的雷射雷达价格是 16 线束的 10 倍。Google 无人车、百度无人车均使用的是高端配置的多线束雷达产品。Velodyne HDL-64E 单个定

制的成本在 8 万元左右，如表 9-2 所示。目前，Velodyne 公司已经开发出了相对便宜的 LiDAR 传感器版本 HDL-32E 和 HDL-16E。其中 HDL-16E 由 16 束雷射取代 64 束雷射，支持 360 度无盲区扫描，牺牲一定的数据规模云点，每秒钟只提供 30 万个数据点，但是售价仍高达 8 千美元，如图 9-4 所示。



图 9-4 Velodyne 激光雷达

表 9-2 Velodyne 激光雷达详细参数^[1]

	HDL-64	HDL-32	VLP-16
价格	8 万美元左右	2 万美元	7999 美元
激光束	64	32	16
扫描范围	120 米	100 米	100 米
精度	正负 2 厘米	正负 2 厘米	正负 2 厘米
数据频率	1.3M 像素/秒	700,000 像素/秒	300,000 像素/秒
角度（垂直/水平）	26.8°/360°	40°/360°	30°/360°
功率	60W	12 W	8 W

激光雷达想在无人车上普及首先应该降低价格。这样就有两种解决办法：其一是采用低线数雷达配合其他传感器，但需搭配拥有极高计算能力系统的无人车；其二是采用固态激光雷达。现今有旋转部件的激光雷达技术较为成熟，国外主流生产厂家为 Velodyne 和 Ibeo。^[2] Velodyne 采用雷射发射、接收一起旋转的方式，产品涵盖 16/32/64 线；Ibeo 采用固定雷射光源，通过内部玻璃片旋转的方式改变雷射光束方向，实现多角度检测，产品涵盖 4/8 线。激光雷达最贵的就是机械旋转部件，固态激光雷达无须旋转部件，采用电子设备替代，因而体积更小，方便集成在车身内部，系统可靠性提高，成本也可大幅降低。由于缺乏旋转部件，水平视角小于 180°，所以需要多个固态雷达组合一起配合使用才行。

在 CES2016 上展出的两款重量级产品，一是来自 Quanergy 的固态（Solid State）雷射雷达 S3，采取相控阵技术，内部不存在任何旋转部件，仅为一盒名片大小，单个售价初步定在 250 美元，量产后可能降至 100 美元；^[3] 二是由 Velodyne 与福特共同发布的混合固态雷射雷达 VLP-16 PUCK，2020 年计划量产价为 500 美元，2025 年计划把成本控制在 200 美元以内。^[4] 奥迪的无人车 A7 Piloted Driving 就采用了 Ibeo 和 Valeo 合作的 Scala 混合固态雷射雷达，在外观上看不到旋转部件，但内部仍靠机械旋转实现雷射扫描。Quanergy、Velodyne、麻省理工学院等都在推进固态激光雷达的研发，其核心在于上游半导体工艺的突破，例如高功率、高波束质量的辐射源、高灵敏度接收技术、产品良率等，如果这些关键指标获得突破，固态激光雷达的实用化有机会让成本下降至 100 美元。

此前国内雷达制造商速腾聚创宣布完成的 16 线激光雷达采用的也是混合固态的形式。固态雷达产品由于采用电子方案去除了机械旋转部件，因此具有低成本（几百美元级别）和体积小、可集成至传统车辆外观中的特点。行业对固态雷达的出现仍处于观望态度，主要因为首先对成本是否能有如此大幅下降抱有疑问；其次，雷射特性在大雾等天气仍然并不适用。

Velodyne 和 Ibeo 的产品规格参数对比如表 9-3 所示。

表 9-3 Velodyne 和 Ibeo 的产品规格参数对比

厂家	产品	价格	激光束	维度	旋转频率	测量参数			分辨率		精度 (CM)
						水平视场 (度)	垂直视场 (度)	探测距离 (米)	水平角 (度)	垂直角 (度)	
Velodyne	HDL-64	8 万 美元	64	3D	5 ~ 20Hz	360	30	100	0.1-0.4	2	3
	HDL-32	2 万 美元	32	3D	5 ~ 20Hz	360	40	100	0.1-0.4	1.33	2
	VLP-16	7999 美元	16	3D	5 ~ 20Hz	360	26.8	120	0.08	0.4	<2
Ibeo	LUX	—	4	2.5D	12.5/25/50Hz	85	3.2	200	0.125	0.8	4
	LUX8	—	8	2.5D	6.5/12.5/25Hz	110	6.4	200	0.125	0.8	10

2. 国内外制造现况

目前，激光雷达已被应用在某些无人驾驶试验车中。

(1) 谷歌和百度的无人驾驶试验车均采用了 Velodyne 的 64 线雷射雷达。

(2) 福特的混动版蒙迪欧安装了 Velodyne 的 32 线雷射雷达，第三代自动驾驶车辆 Fusion Hybrid 配置了 2 台 Velodyne 的混合固态雷射雷达。

(3) 日产 LEAF 搭载了 6 个 Ibeo 的 4 线雷射雷达，测试了其高级驾驶辅助系统。

(4) 奥迪的无人车 A7 Piloted Driving 采用了 Ibeo 和 Valeo 合作的 Scala 混合固态雷射雷达。

(5) 德尔福无人车配备了 4 台由 Quanergy 研发的固态雷射雷达。

(6) 2015 年，奥迪的无人车 A7 Piloted Driving 就采用了 Ibeo 和 Valeo 合作的 Scala 混合固态雷射雷达，大众的一款半自动驾驶汽车也搭载了 Scala，该雷射雷达隐藏在保险杠内，用于取代毫米波雷达做 AEB 的测距模块。

国外激光雷达研发厂商比较有代表性的有 Velodyne、Ibeo 和 Quanergy，并且他们都背靠巨头。Velodyne 成立于 1983 年，位于加州硅谷。当年，美国举办的世界无人车挑战赛获得第一名和第二名的高校卡耐基梅隆大学和斯坦福大学使用的就是 Velodyne 的激光雷达。目前，其已有包括 Velodyne 16、32 和 64 线激光雷达三个系列。Ibeo 是无人驾驶激光雷达供应商，成立于 1998 年，2010 年和法雷奥合作开始量产可用于汽车的产品 Scala，其目前主要供应 4 线和 8 线的激光雷达。位于加州硅谷中心的 Quanergy 成立于 2012 年，虽然相对其他激光雷达厂家较为年轻，但它却制造出了全球第一款固态激光雷达。

国内在激光雷达研发这块的企业主要有北醒光子、思岚科技、镭神智能、速腾聚创和禾赛科技。其中北醒光子目前的产品有三大系列：单线环境雷达 DE-LiDAR 1.0、多线长距雷达 DE-LiDAR 2.0 和固态雷达 DE3.0 系列（多线长距雷达目前正在研发，可做到 8 到 32 线）；^[5] 镭神智能成立于 2015 年初，是一家提供中远距离脉冲测距激光雷达等产品及解决方案的公司；^[6] 速腾聚创已宣布完成其混合固态的 16 线激光雷达研发。^[7]

在无人驾驶领域，激光雷达是目前最有效的方案，被认为是最精准的自主感知手段，其有效感知范围超过 120 米，而精度可以达到厘米级，也是其中最重要的目前最难以跨越的硬件门槛。由于现在价格高昂，无法部署在量产车上，多线激光雷达的成本下降将加速

无人驾驶的落地，如果在价格上也相当，前景无疑是乐观的，相信无人驾驶离我们就不远了。

9.2.2 毫米波雷达

毫米波雷达通过发射无线电信号（毫米波波段的电磁波）并接收反射信号来测定汽车车身周围的物理环境信息（如汽车与其他物体之间的相对距离、相对速度、角度、运动方向等），然后根据所探知的物体信息进行目标追踪和识别分类，进而结合车身动态信息进行数据融合，完成合理决策，减少事故发生概率。

毫米波雷达的工作频段为 30GHz~300GHz，毫米波的波长为 1~10mm，介于厘米波和光波之间，因此毫米波兼有微波制导和光电制导的优点。雷达测量的是反射信号的频率转变，并计算其速度变化。雷达可以检测 30~100m 远的物体，高端的雷达能够检测到很远的物体。同时，毫米波雷达不受天气状况限制，即使是雨雪天也能正常运作，穿透雾、烟、灰尘的能力强。具有全天候、全天时的工作特性，且探测距离远，探测精度高，被广泛应用于车载距离探测，如自适应巡航、碰撞预警、盲区探测等，如图 9-5 所示。

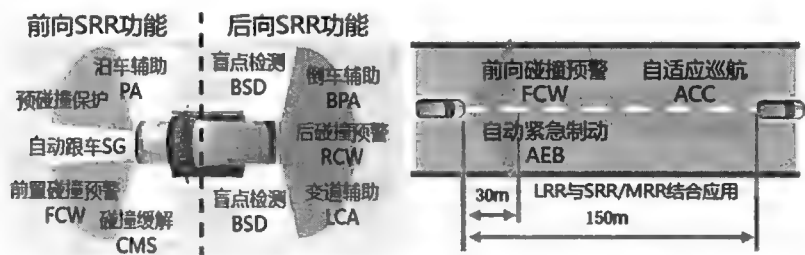


图 9-5 ADAS 中毫米波雷达应用范围原理示意

相比激光雷达，毫米波雷达精度低、可视范围的角度也偏小，一般需要多个雷达组合使用。雷达传输的是电磁波信号，因此它无法检测上过漆的木头或是塑料（隐形战斗机就是通过表面喷漆躲过雷达信号的），行人的反射波较弱几乎对雷达免疫。同时，雷达对金属表面非常敏感，如果是一个弯曲的金属表面，它会被雷达误认为是一个大型表面。因此，路上一个小小的易拉罐甚至可能被雷达判断为巨大的路障。此外，雷达在大桥和隧道里的效果同样不佳。

1. 毫米波雷达分类

毫米波雷达的可用频段有 24GHz、60GHz、77GHz 和 79GHz，主流可用频段为 24 GHz 和 77GHz，分别应用于中短距和中长距测量，如图 9-6 所示。相比于 24GHz，77GHz 毫米波雷达物体分辨准确度可提高 2~4 倍，测速和测距精确度提高 3~5 倍，能检测行人和自行车，且设备体积更小，更便于在车辆上安装和部署。如表 9-4 所示，长距离雷达的侦测范围更广，可适配行驶速度更快的车辆，但是相应地探测精度下降，因此更适用于 ACC 自适应巡航这类的应用。典型的长距离雷达有博世的一款产品，其探测前向距离为 250 米；典型的短距离雷达有大陆的一款产品，其探测距离为前向 60 米后向 20 米。

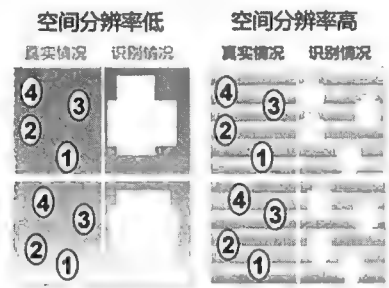


图 9-6 中距雷达和短距雷达空间分辨率对比

表 9-4 中长距雷达和短距雷达参数对比^[8]

	LRR 长距离雷达	SRR/MRR 短距离雷达
分类	窄带雷达	宽带雷达
覆盖距离（m）	280	30/120
车速上限 km/h	250	150
精度	0.5m	厘米级
主要应用范围	ACC 自适应巡航	车辆环境监测

为完全实现 ADAS 的各项功能一般需要“1 长+4 中短”5 个毫米波雷达，目前全新奥迪 A4 采用的就是“1 长+4 短”5 个毫米波雷达的配置。以自动跟车型 ACC 功能为例，一般需要 3 个毫米波雷达。车正中间一个 77GHz 的 LRR，探测距离在 150~250m，角度为 10 度左右；车两侧各一个 24GHz 的 MRR，角度都为 30 度，探测距离在 50~70m。图 9-7 所示为奔驰的 S 级车型，采用的是 7 个毫米波雷达（1 长+6 短）。

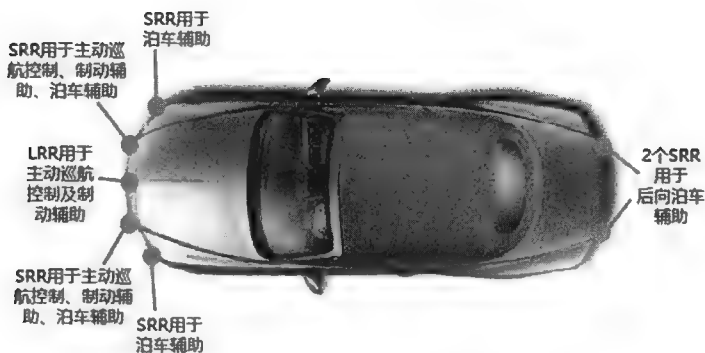


图 9-7 毫米波雷达在无人驾驶中的应用示意图

电磁波频率越高，距离和速度的检测解析度越高，因此频段发展趋势是逐渐由 24GHz 向 77GHz 过渡的。1997 年，欧洲电信标准学会确认 76~77GHz 作为防撞雷达专用频道。早在 2005 年，原信息产业部发布《微功率（短距离）无线电设备的技术要求》，将 77GHz 划分给车辆测距雷达。2012 年，工信部进一步将 24GHz 划分给短距车载雷达业务。2015 年，日内瓦世界无线电通信大会将 77.5~78.0GHz 频段划分给无线电定位业务，以支持短距离高分辨率车载雷达的发展，从而使 76~81GHz 都可用于车载雷达，为全球车载毫米波雷达的频率统一指明了方向。至此之后，最终车载毫米波雷达将会统一于 77GHz 频段（76~81GHz），该频段带宽更大、功率水平更高、探测距离更远。

2. 毫米波雷达国内外制造现况

全球汽车毫米波雷达主要供应商为传统汽车电子优势企业，如博世、大陆、Hella、富士通天、电装、TRW、德尔福、Autoliv、法雷奥等传统优势企业，如图 9-8 所示。



图 9-8 毫米波雷达主要供应商

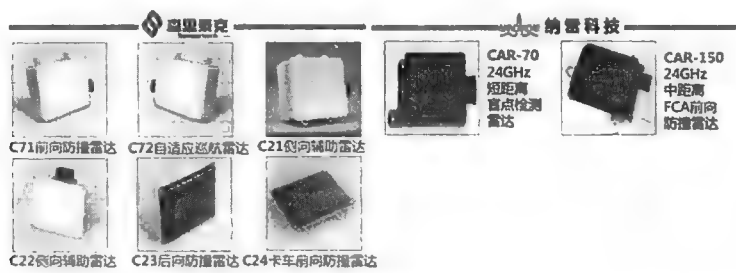
各个主要厂商的主要毫米波雷达产品如表 9-5 所示。

表 9-5 各个主要厂商主要毫米波雷达产品

厂商	长 距		中 距		短 距	
	型号	性能参数	型号	性能参数	型号	性能参数
BOSCH	远距 LRR4	76~77GHz, 前向距离为 250m	中距 MRR	76~77GHz, 前向距离为 160m/42°, 后向距离为 80m/150°	—	—
continental	长距 ARS410	76~77GHz, 前向距离为 170m	—	—	SRR320	24~25GHz
	长距 ARS430	76~77GHz, 前向距离为 250m	—	—	—	—
HELLA	—	—	—	—	短距离雷达 SRR	24GHz, 前向距离为 0.75~70m, 视角 165°
Delphi	—	—	中距 ESR2.5	76~77GHz, 前向距离为 174m	—	—
Denso	长距离雷达 LRR	76~77GHz, 前向距离为 205m, 视角 36°	—	—	—	—
Autoliv	—	—	—	—	短距离雷达 SRR 25GHz	超宽带 24GHz, 窄带 77GHz 多模式雷达

其中，博世的核⼼产品是长距离毫米波雷达，主要用于 ACC 系统。最新产品 LRR4 可以探测 250 米外的车辆，是目前探测距离最远的毫米波雷达，市场占有率最高，但客户集中在奥迪和大众。大陆客户分布广，产品线齐全，主力产品为 24GHz 毫米波雷达，并且在 Stop & Go ACC 领域占有率极高。Hella 在 24GHz-ISM 领域客户范围最广，24GHz 雷达传感器下线 1000 万片，出货量达 650 万片，市场占有率全球第一。第四代 24GHz 雷达传感器将在 2017 年中投入全球化生产。富士通天和电装主要占据日本市场，其中富士通天略胜一筹。富士通天、松下和电装是未来 79GHz 雷达市场领域的强者。从工艺上看，毫米波雷达正从点目标探测往成像雷达⽅面发展，例如 SAR（相控阵）型雷达，正在从军用领域向汽车领域推进。

目前，中国市场中⼾端汽车装配的毫米波雷达传感器全部依赖进口，国内⾃主车⾃毫米波雷达产品总体仍处于研制阶段。因研发成本及难度较低，国际市场上 24GHz 毫米波雷达供应链也已相对稳定，目前国内厂商的研发⽅向主要集中于 24GHz 雷达产品，可从飞思卡尔等供应商获得 24GHz 射频芯片。目前，较为成熟的产品仅有湖南纳雷、厦⼾意行、芜湖森思泰克的 24GHz 中短距雷达，⽽ 77GHz 产品设计难度较大，成本较高，并且英飞凌、ST、飞思卡尔等芯片厂商并没有对中国开放供应 77GHz 射频芯片，因此国内 77GHz 毫米波雷达的开发受到很⼾限制，如图 9-9 所示。



片也正在从硅锗工艺向廉价的 CMOS 工艺发展,预计在 2020 年左右,COMS 工艺可以达到现有硅锗工艺的水平,并且发射、接收及信号处理器三合一的产品也在开发中,届时,毫米波雷达芯片将可能比目前的价格下降数倍。

9.2.3 车载摄像头

车载摄像头的大致原理是:首先,采集图像进行处理,将图片转换为二维数据;然后,进行模式识别,通过图像匹配进行识别,如识别车辆行驶环境中的车辆、行人、车道线、交通标志等;接下来,依据物体的运动模式或使用双目定位,以估算目标物体与本车的相对距离和相对速度。

相比于其他传感器,摄像头是最接近人眼获取周围环境信息的工作模式,可以通过较小的数据量获得最全面的信息。同时,因为现在的摄像头技术比较成熟,成本可较低。但是,摄像头识别也存在一定局限性,基于视觉的解决方案受光线、天气影响大。而且,物体识别基于机器学习资料库,需要的训练样本大,训练周期长,也难以识别非标准障碍物。此外,由于广角摄像头的边缘畸变,得到的距离准确度较低。

从应用方案出发,目前摄像头可划分为单目、后视、立体(双目)和环视摄像头四种,如表 9-6 所示。

表 9-6 各摄像头传感器的应用场景

	应用场景
单目摄像头	ACC、LDW、LKA、FCW、AEB、TSR、AP、PDS、DMS
后视摄像头	AP
立体(双目)摄像头	ACC、LDW、LKA、FCW、AEB、TSR、AP、PDS、DMS
环视摄像头	AP、SVC

(1) 单目摄像头一般安装在前挡风玻璃上部,用于探测车辆前方环境、识别道路、车辆、行人等。先通过图像匹配进行目标识别(各种车型、行人、物体等),再通过目标在图像中的大小估算目标距离。这要求对目标进行准确识别,然后建立并不断维护一个庞大的样本特征数据库,保证这个数据库包含待识别目标的全部特征数据。如果缺乏待识别目标的特征数据,就无法估算目标的距离,导致 ADAS 系统的漏报。因此,单目视觉方案的技术难点在于模型机器学习的智能程度或者模式识别的精度。

(2) 后视摄像头,一般安装在车尾,用于探测车辆后方环境,技术难点在于如何适应

不同的恶劣环境。

(3) 立体(双目)摄像头是通过两幅图像视差的计算,直接对前方景物(图像所拍摄到的范围)进行距离测量,而无须判断前方出现的是什么类型的障碍物。依靠两个平行布置的摄像头产生的“视差”,找到同一个物体所有的点,依赖精确的三角测距,就能够算出摄像头与前方障碍物的距离,实现更高的识别精度和更远的探测范围。使用这种方案,需要两个摄像头有较高的同步率和采样率,因此技术难点在于双目标定及双目定位。相比单目,双目的解决方案没有识别率的限制,无须先识别,可直接进行测量;直接利用视差计算距离精度更高;无须维护样本数据库。因为检测原理上的差异,双目视觉方案在距离测算上相比单目及毫米波雷达、激光雷达,其硬件成本和计算量级的加倍也是另一个难关。

(4) 环视摄像头,一般至少包括四个摄像头,分别安装在车辆前、后、左、右侧,实现 360° 环境感知,难点在于畸变还原与对接,如图 9-10 所示。

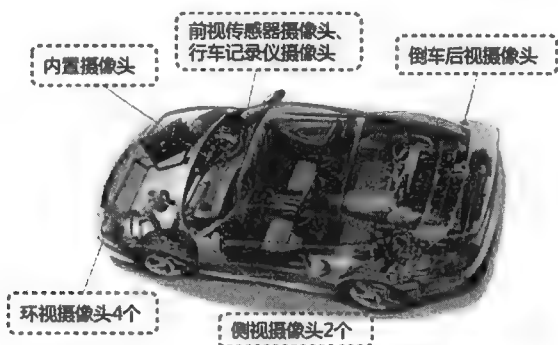


图 9-10 无人车中各摄像头传感器的方位设置

根据不同 ADAS 功能的需要,摄像头的安装位置也有不同。主要分为前视、后视、侧视以及内置,如表 9-7 所示。实现(半)自动驾驶时全套 ADAS 功能将安装 6 个以上摄像头。

表 9-7 按功能需求的摄像头传感器划分

安装部位	摄像头类型	应用场景
前视	单目、双目	FCW、LDW、TSR、ACC、PCW
环视	广角	全景泊车、LDW
后视	广角	后视泊车辅助

续表

安装部位	摄像头类型	应用场景
侧视	广角	盲眼检测、替代后视镜
内置	广角	闭眼提醒

如图 9-11 所示，前视摄像头一般采用 55 度左右的镜头得到较远的有效距离，有单目和双目两种解决方案。双目需要装在两个位置，成本较单目贵 50%。环视使用的是广角摄像头，通常在车四周装备四个进行图像拼接实现全景图，通过辅助算法可实现道路线感知。后视采用广角或者鱼镜头，主要为倒车后视使用。侧视一般使用两个广角摄像头，完成盲点检测等工作，也可代替后视镜，这一部分功能也可由超声波雷达替代。内置使用的也是广角镜头，安装在车内后视镜处，完成在行驶过程中对驾驶员的闭眼提醒。其中，前视摄像头可以实现车道偏离预警、车辆识别应用、车辆识别、行人识别、道路标识识别等 ADAS 主动安全的核心功能，未来将成为自动紧急刹车（AEB）、自适应巡航（ACC）等主动控制功能的信号入口。这种摄像头安全等级较高，应用范围较广，是目前开发的热点。



图 9-11 各类无人驾驶中广泛应用的摄像头传感器^[9]

车载摄像头在工艺上具有的首要特性是快速，特别是在高速行驶场合，系统必须能记录关键驾驶状况、评估这种状况并实时启动相应措施。在 140 km/h 的速度下，汽车每秒要移动 40 公尺。为避免两次图像信息获取间隔期间自动驾驶的距离过长，要求相机具有最慢不低于 30 帧/秒的影像捕捉速率，在汽车制造商的规格中，甚至提出了 60 帧/秒和 120 帧/秒的要求。在功能上，车载摄像头需要在复杂的运动路况环境下保证采集到稳定的数据。具体表现如下。

- （1）高动态：在较暗环境及明暗差异较大时仍能实现识别，要求摄像头具有高动态的特性。
- （2）中低像素：为降低计算处理的负担，摄像头的像素并不需要非常高。目前 30 万~120 万像素已经能满足要求。

(3) 角度要求: 对于环视和后视, 一般采用 135 度以上的广角镜头, 前置摄像头对视距要求更大, 一般采用 55 度的范围。

同时, 相比工业级与生活级摄像头, 车载类型在安全级别上要求更高, 尤其是对前置 ADAS 的镜头安全等级要求更高。主要体现如下。

(1) 温度要求: 车载摄像头温度范围在 -40 度~80 度。

(2) 防磁抗振: 汽车启动时会产生极高的电磁脉冲, 车载摄像头必须具备极高的防磁抗振的可靠性。

(3) 较长的寿命: 车载摄像头的寿命至少要在 8~10 年以上才能满足要求。

根据 IHS Automotive 的预测, 车载摄像头系统出货量有望在 2021 年达到 7400 万套/年。国内行业龙头优势地位明显, 如舜宇光学车载后视镜镜头出货量目前居全球第 1 位, 全球市场占有率达 30% 左右, 产品包括前视镜头、后视镜镜头、环视镜头、侧视镜头、内视镜头等。客户遍及欧美、日韩和国内, 广泛应用于 BMW、Benz、Audi、Toyota、Honda、Chrysler、Ford、GM、VW、Volvo 等众多车型上, 具体的型号包括 4005、4408、4009、4017、4017、4034、4043、4044 等。以 4005 与 4043 为例, 其规格参数如表 9-8 所示。

表 9-8 舜宇光学 4005、4043 视觉传感器规格参数

型号	ELF (mm)	HFOV (°)	Max Image Circle	Resolution
4005	1.02	138	1/4" (Φ5.0)	VGA
4043	1.1	187	H1/4" (Φ4.0)	MEGA

9.2.4 GPS/IMU

GPS 在复杂的动态环境中, 尤其在大城市, 其多路径反射的问题会很显著, 导致获得的 GPS 定位信息很容易产生几米的误差。另外, 由于 GPS 的更新频率低 (10Hz), 在车辆快速行驶时很难给出精准的实时定位。单纯依赖 GPS 的导航很有可能导致交通事故。因此, GPS 通常辅助以惯性传感器 (IMU) 用来增强定位的精度。IMU 是检测加速度与旋转运动的高频 (1KHz) 传感器, 但 IMU 自身也有偏差积累与噪音等问题影响结果。通过使用基于卡尔曼滤波的传感器融合技术, 我们可以融合 GPS 与 IMU 数据, 结合 GPS 的定位精度高和误差无积累的特点, 与 IMU 的自主性和实时性的优点。一方面可以实现导航设备之间优势互补, 增强系统适应动态的能力, 并使整个系统获得优于局部系统的精度; 另一方面提高了空间和时间的覆盖范围, 从而实现真正意义上的连续导航。因此,

GPS/IMU 组合的优势有以下几点。

1. 系统精度的提高

利用 GPS 的长期稳定性弥补 IMU 误差随时间累积的缺点。GPS/IMU 组合后的导航误差实际上要比单独的 GPS 或单独的惯导系统可能达到的误差都小。

2. 系统抗干扰能力的增强

利用 IMU 的短期高精度弥补 GPS 系统易受干扰、信号易失锁等缺点，同时借助 IMU 的姿态信息、角速度信息可进一步提高 GPS 系统快速捕获或重新锁定卫星信号的能力。

3. 导航信息的补全

GPS/IMU 组合系统与单 GPS 相比，除了可以提供载体运动的三维位置和速度信息外，还可提供加速度、姿态和航向信息；GPS/IMU 组合系统可提供 100Hz 甚至高于 100Hz 的数据更新率。

IMU 惯性器件的标定技术由于加速度计、陀螺仪等惯性器件本身存在缺陷，会产生一些器件误差，如标度因数误差等。另外，在对 IMU 进行集成时，各个器件之间的非正交安装会引起交叉耦合误差。以上这些误差可以通过器件标定加以补偿，以达到提高其精度的目的。

GPS/IMU 的主要制造商有 NovAtel、Leica、CSI Wireless 及 Thales Navigation。其中，NovAtel 提出了 SPAN 技术。SPAN 集合了 GPS 定位的绝对精度与 IMU 陀螺和加速计测量的稳定性，以提供一个 3D 的位置、速度和姿态解算结果。即使在 GPS 信号被遮挡时，也能提供稳定连续的解算结果。如图 9-12 所示，基于 SPAN 技术，NovAte 有两款主要的 GPS/IMU 产品：SPAN-CPT 一体式组合导航系统与 SPAN-FSAS 分式组合导航系统。SPAN-CPT 采用 NovAtel 自主的专业级的高精度 GPS 板卡与德国的 iMAR 公司制造的光纤陀螺 IMU，其解算精度在不同的模式下可适用于不同的定位需求，支持包括 SBAS、L 波段（Omnistar 和 CDGPS）和 RTK 差分等多种方式、系统最高航向精度 0.05°、俯仰横滚精度 0.015°。SPAN-FSAS 也采用德国 iMAR 公司高精度、闭环技术的 IMU，其陀螺偏差小于 0.75 度/小时和加速计偏差小于 1mg，配合目前 NovAtel 的 FlexPak6™或 ProPak6™集成了组合导航解算。从 IMU-FSAS 的惯性测量数据发送到 GNSS 接收机进行解算，GNSS + INS 的位置、速度和姿态输出速率高达 200Hz。

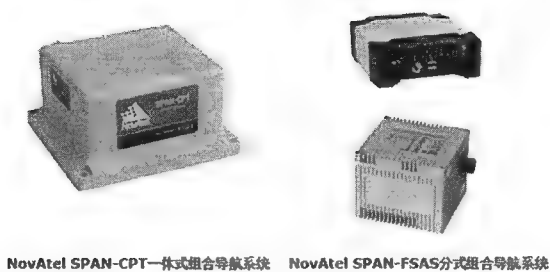


图 9-12 NovAtel 的两款 GPS/IMU 产品图

9.2.5 V2X 通信传感

V2X 通信系统可以看作是一个超级传感器，它提供了比其他车载传感器都高得多的感知能力和可靠性，在自车感知技术尚不能达到高可靠性之前，用 V2X 可以决定性地提升其可靠性。V2X 是无人驾驶必要技术和智慧交通的重要一环。V2X 是 V2V（Vehicle to Vehicle，车车通信）、V2I（Vehicle to Instruction，车路通信）、V2P（Vehicle to Pedestrian，车人通信）等的统称，通过 V2X 可以获得实时路况、道路信息、行人信息等一系列交通信息，从而带来远距离环境信号。简单来说，V2V 技术是利用无线通信技术实现车与车之间、车与道路之间、车与行人之间的信息互通，也就是说，通过人、车、路之间的相互交流，使驾驶员能更好地掌握车辆状态和周围情况，驾驶员收到警告后就能降低事故的风险或车辆本身就会采取自治措施，像是制动减速。V2V 通信技术首次由福特公司在 2014 年 6 月 3 日发布，在现场展示的是福特的两辆经过特殊改造过的车，通过一台连接了 Wi-Fi 的无线广播系统，演示了这项 V2V 通信技术是如何防止碰撞事故发生的，如图 9-13 所示。

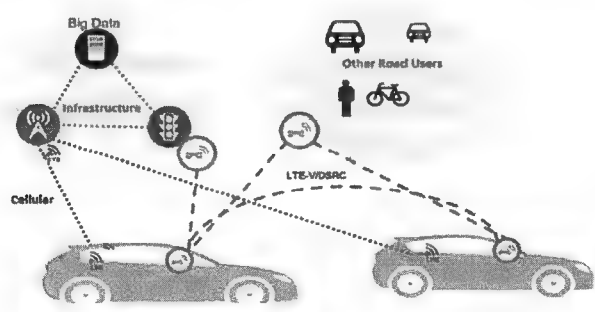


图 9-13 V2X 通信辅助下的行车状况示意图

详细来说，V2X 是一种网状网络，网络中的节点（汽车、智能交通灯等）可以发射、

捕获并转发信号。利用 V2X 车联网,车辆可以获取周围环境的未知参数及附近车辆的运行状态,这些状态包括速度、位置、驾驶方向、刹车等基本的安全信息。然后车载端主动安全算法将处理所获取的信息,并按照优先级对信息进行分类,对可能发生的危险情景进行预警,紧急情况下可以利用车辆执行端对车辆进行控制从而规避风险。V2X 技术开启了对四周威胁的 360 度智能感知,这一技术能够在各种危险情况下提醒驾驶者,从而大大减少汽车碰撞事故的发生并缓解交通拥堵。美国交通部根据最新的数据进行分析,如果能够大面积地普及 V2X 技术,就能在 75% 的交通事故发生之前提醒驾驶员。

1. V2X 通信优势

相比传统雷达, V2X 通信传感系统有以下几点优势。

1) 覆盖面更广

300~500 米的通信范围相比十几米的雷达探测范围要远得多,不仅是前方障碍物,身旁和身后的建筑物、车辆都会互相连接,大大拓展了驾驶员的视野范围,驾驶员能获得的信息也就更多也更立体。因此,在前车刹车初期就能有效甄别,并进行提示,如果距离过近,系统会再次提示,对预判和规避危险也有足够的反应时间,避免出现跟车追尾的情况。

2) 有效避免盲区

由于所有物体都接入互联网,每个物体都会有单独的信号显示,因此即便是视野受阻,通过实时发送的信号可以显示视野范围内看不到的物体状态,也就降低了盲区出现的概率,也就充分避免了因盲区而导致的潜在伤害。

3) 对于隐私信息的安全保护性更好

由于这套系统将采用 5.9Hz 频段进行专项通信,相比传统通信技术更能确保安全性和私密性,如果通信协议及频道在各个国家都能够规范化,这套系统将变得像 SOS 救援频道一样成为社会公用资源。

2. V2X 通信的国内外发展进展

1) 国外 V2X 进展

目前,这套 V2V 协议由通用、福特、克莱斯勒等厂商联合研发,除了美国汽车三巨头,丰田、日产、现代、起亚、大众、奔驰、马自达、斯巴鲁、菲亚特等车企也在协议名单内。2016 年 12 月 14 日,美国交通部发布了 V2V 的新法规,进入了 90 天公示期,法

规强制要求新生产的轻型汽车安装 V2V 通信装置，这是一个里程碑式的进步。V2V 车企联盟如图 9-14 所示。



图 9-14 V2V 车企联盟

美国交通部的新规中要求 V2V 装置的通信距离达到 300 米，并且是 360 度覆盖，远超摄像头的探测能力，其感知信息属于结构化信息，不存在误报的可能。根据美国国家公路交通安全管理局（NHTSA）的研究，利用 V2X 技术，可以减少 80% 的非伤亡事故，但这一切是以 100% 的覆盖率为前提的。在此之前，如凯迪拉克等车企也曾经做过尝试但都因缺乏足够的覆盖率难以发挥作用，依靠强制性的法规驱动，V2X 普及的最大难题将得以有效解决。

高通发布新闻表示，将与奥迪、爱立信等公司进行蜂窝-V2X（Celluar-V2X）的测试合作，该测试符合由德国政府主导的项目组织——自动互联驾驶数字测试场的测试规范。在此之前，高通推出了基于其最新骁龙 X16 LTE modem 的全新联网汽车参考平台，支持作为可选特性的专用短程通信（DSRC）和蜂窝-V2X。

2) 中国 V2X 的发展进展

2016 年下半年，发改委连同交通部联合发布了《推进“互联网+”便捷交通促进智能交通发展的实施方案》^[10]，明确提出“结合技术攻关和试验应用情况，推进制定人车路协同国家通信标准和设施设备接口规范，并开展专用无线频段分配工作”的标准制定工作。从目前的情况来看，LTE-V 极有可能被确定为中国标准。5G 的推进对 V2X 是非常大的利好，因为 5G 标准本身就包含了 V2X，可以说 5G 的发展和无人驾驶的发展是自洽的。

为了满足在商业应用上的高可靠性，越来越多的车企意识到在增强车辆能力的同时，需要将道路从对人友好改造为对车友好，从 2015 开始，中国所有的无人驾驶示范园区都在规划部署路侧系统（V2I）。随着 5G 的时间表日渐清晰，更大范围的部署也让人非常期待。5G 的核心推动力来自物联网，而汽车可能是其中最大的单一应用，一辆无人车每天可以产生超过 1TB 的数据。目前，多个地图供应商正在积极准备用于无人驾驶的实时高

精地图，以克服静态高精地图无法适应道路变化的难题，但之前受制于无线带宽限制，很难达到实用，5G 可提供高达 10Gbit/s+的峰值速率，以及 1ms 的低延时性能，可以满足这样的需求。

9.2.6 传感器小结

各种传感器的比较如表 9-9 所示。

表 9-9 各种传感器的比较

传感器	成 本	优 势	劣 势	功 能
激光雷达	8000 美元以上	扫描周围环境得到精确环境信息	成本高，大雾、雨雪天气效果差，无法图像识别	周边环境 3D 建模
毫米波雷达	300~500 美元	不受天气影响，测量精度高，距离范围广	无法识别道路指示牌，无法识别行人	无法应用视觉识别要求较高功能
摄像头	35~50 美元	成本比较低，通过算法可以实现各种功能	极端恶劣环境下会失效，难以测距，距离较近，算法要求高	能实现大多数 ADAS 功能，测距功能难以实现
V2X	150~200 美元	不受距离现实，V2X 成本较低，深度融合智能系统	精度较低，技术协议仍在讨论中，普及难度大	利用通信协议，感知实时路况，道路信息和行人信息
红外传感器	600~2000 美元	夜视效果极佳	成本较高，技术仍由国外垄断	夜视
超声波传感器	15~20 美元	成本低	探测距离较近，应用局限大	侧方超车提醒、倒车提醒

9.3 计算平台

当硬件传感器接收到环境信息后，数据会被导入计算平台，由不同的芯片进行运算。计算平台的设计直接影响到无人驾驶系统的实时性及鲁棒性。本章将深入了解无人驾驶计算平台。

9.3.1 计算平台实现

为了了解无人驾驶计算平台的要点，我们来看看一个行业领先的某四级无人驾驶公司

现有的计算平台硬件实现。为了了解芯片制造商将如何解决这些问题，我们来看看现有的不同芯片制造商所提供的无人驾驶计算解决方案。

这个四级无人驾驶公司的计算平台由两个计算盒组成。每个计算盒配备了一颗英特尔至强 E5 处理器（12 核）和 4 到 8 颗 NVIDIA K80 GPU 加速器，彼此使用 PCI-E 总线连接。CPU 运算峰值速度可达 400 帧/秒，功率需求 400W。每个 GPU 运算峰值速度可达 8 TOP/S，功率需求 300W。因此，整个系统能够提供 64.5 TOP/S 的峰值运算能力，其功率需求为 3000 W。计算盒与车辆上安装的 12 个高精度摄像头相连接，以完成实时的物体检测和目标跟踪任务。车辆顶部还安装有一个激光雷达装置以完成车辆定位及避障功能。为了保证可靠性，两个计算盒执行完全相同的任务。一旦第一个计算盒失效，第二个计算盒可以立即接管。在最坏的情况下，两个计算盒都在计算峰值运行，这意味着将产生超过 5000W 的功耗并急聚大量的热量，散热问题不容忽视。此外，每个计算盒的成本预计为 2 万~3 万美元，这是普通消费者根本无法承受的整体解决方案。

9.3.2 现有的计算解决方案

本节，我们将分别介绍现有的针对无人驾驶的计算解决方案。

1. 基于 GPU 的计算解决方案

GPU 在浮点运算、并行计算等部分的计算方面能够提供数十倍至上百倍的 CPU 性能。利用 GPU 运行机器学习模型，在云端进行分类和检测，其相对于 CPU 耗费的时间大幅缩短，占用的数据中心的基础设施更少，能够支持比单纯使用 CPU 时 10~100 倍的应用吞吐量。凭借强大的计算能力，在机器学习快速发展的推动下，GPU 目前在深度学习芯片市场非常受欢迎，很多汽车生产商也在使用 GPU 作为传感器芯片发展无人车，GPU 大有成为主流的趋势。研究公司 Tractica LLC 预计，到 2024 年深度学习项目在 GPU 上的花费将从 2015 年的 4360 万美元增长到 41 亿美元，在相关软件上的花费将从 1.09 亿美元增长到 104 亿美元。

凭借具备识别、标记功能的图像处理器，在人工智能还未全面兴起之前，NVIDIA 就先一步掌控了这一时机。在 2016 年，NVIDIA 更是一连发布了多款针对深度学习的芯片，如 4 月份发布的一款可执行深度学习神经网络任务的 Tesla P100 GPU，又如 9 月份发布的基于 Pascal 架构的深度学习芯片 Tesla P4 和 Tesla P40，其中，Pascal 架构能助推深度学习加速 65 倍。

NVIDIA 的 PX2 平台是目前领先的基于 GPU 的无人驾驶解决方案。每个 PX2 由两个 Tegra SoC 和两个 Pascal GPU 图形处理器组成,其中每个图像处理器都有自己的专用内存并配备有专用的指令以完成深度神经网络的加速。为了提供高吞吐量,每个 Tegra SOC 使用 PCI-E Gen 2 x4 总线与 Pascal GPU 直接相连,其总带宽为 4 GB/s。此外,两个 CPU-GPU 集群通过千兆以太网相连,数据传输速度可达 70 Gigabit/s。借助优化的 I/O 架构与深度神经网络的硬件加速,每个 PX2 能够每秒执行 24 兆次深度学习计算。这意味着当运行 AlexNet 深度学习典型应用时, PX2 的处理能力可达 2800 帧/秒,如图 9-15 所示。^[11]

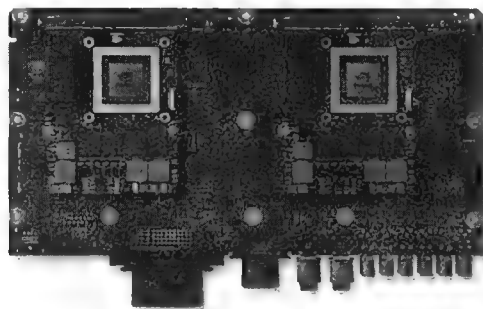


图 9-15 NVIDIA PX2 平台芯片示意图

2. 基于 DSP 的解决方案

DSP (Digital Singnal Processor) 以数字信号处理大量数据。DSP 采用的是哈佛设计,即数据总线和地址总线分开,允许取出指令和执行指令完全重叠,在执行上一条指令的同时就可取出下一条指令,并进行译码,这大大提高了微处理器的速度。另外,还允许在程序空间和数据空间之间进行传输,因为增加了器件的灵活性。它不仅具有可编程性,而且其实时运行速度可达每秒数以千万条复杂指令程序,远远超过通用微处理器。它的强大数据处理能力和高运行速度是最值得称道的两大特色。由于它的运算能力很强,速度很快,体积很小,而且采用软件编程具有高度的灵活性,因此为从事各种复杂的应用提供了一条有效途径。

德州仪器提供了一种基于 DSP 的无人驾驶的解决方案。其 TDA2x SoC 拥有两个浮点 DSP 内核 C66x 和四个专为视觉处理设计的完全可编程的视觉加速器。相比 ARM Cortex-15 处理器,视觉加速器可提供 8 倍的视觉处理加速且功耗更低。^[12]类似的设计有 CEVA XM4。这是另一款基于 DSP 的无人驾驶计算解决方案,专门面向计算视觉任务中的视频流分析计算。使用 CEVA XM4 每秒处理 30 帧 1080p 的视频仅消耗功率 30mw,是

一种相对节能的解决方案，如图 9-16 所示。^[13]

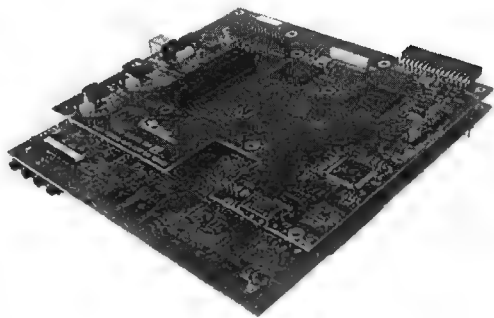


图 9-16 TI TDA2 SoC 芯片示意图

3. 基于 FPGA 的解决方案

作为 GPU 在算法加速上强有力的竞争者，FPGA 硬件配置最灵活，具有低能耗、高性能及可编程等特性，十分适合感知计算。更重要的是，FPGA 相比 GPU 价格便宜（虽然性价比不一定是最好的）。在能源受限的情况下，FPGA 相对于 CPU 与 GPU 有明显的性能与能耗优势。FPGA 低能耗的特点很适合用于传感器的数据预处理工作。此外，感知算法不断发展意味着感知处理器需要不断更新，FPGA 具有硬件可升级、可迭代的优势。使用 FPGA 需要具有硬件的知识，对许多开发者有一定难度，因此 FPGA 也常被视为一种行家专属的架构。不过，现在也出现了用软件平台编程 FPGA，弱化了软硬件语言间的障碍，让更多开发者使用 FPGA 成为可能。随着 FPGA 与传感器结合方案的快速普及，视觉、语音、深度学习的算法在 FPGA 上进一步优化，FPGA 极有可能逐渐取代 GPU 与 CPU 成为无人车、机器人等感知领域上的主要芯片。

譬如百度的机器学习硬件系统就是用 FPGA 打造了 AI 专有芯片，制成了 AI 专有芯片版百度大脑——FPGA 版百度大脑。在百度的深度学习应用中，FPGA 相比相同性能水平的硬件系统消耗能率更低，将其安装在刀片式服务器上，可以完全由主板上的 PCI Express 总线供电，并且使用 FPGA 可以将一个计算得到的结果直接反馈到下一个，不需要临时保存在主存储器，所以存储带宽要求也在相应降低。

Altera 公司推出的 Cyclone V SoC 是一个基于 FPGA 的无人驾驶解决方案，现已应用在奥迪无人车产品中。Altera 公司的 FPGA 专为传感器融合提供优化，可结合分析来自多个传感器的数据完成高度可靠的物体检测。^[14]类似的产品有 Zynq 专为无人驾驶设计的 Ultra ScaleMPSoC。当运行卷积神经网络计算任务时，Ultra ScaleMPSoC 运算效能为 14

帧/秒/瓦，优于 NVIDIA Tesla K40 GPU 可达的 4 帧/秒/瓦。同时，在目标跟踪计算方面，Ultra ScaleMPSoC 在 1080p 视频流上的处理能力可达 60 fps，如图 9-17 所示。

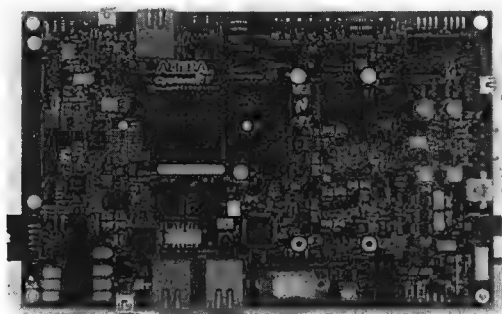


图 9-17 Altera Cyclone V SoC 芯片示意图

4. 基于 ASIC 的解决方案

MobilEye 是一家领先的基于 ASIC 的无人驾驶解决方案提供商。其 EyeQ5 SOC 装备有四种异构的全编程加速器，分别对专有的算法进行了优化，包括计算机视觉、信号处理和机器学习等。EyeQ5 SOC 同时实现了两个 PCI-E 端口以支持多处理器间通信。这种加速器架构尝试为每一个计算任务适配最合适的计算单元，硬件资源的多样性使应用程序能够节省计算时间并提高计算效能，如图 9-18 所示。

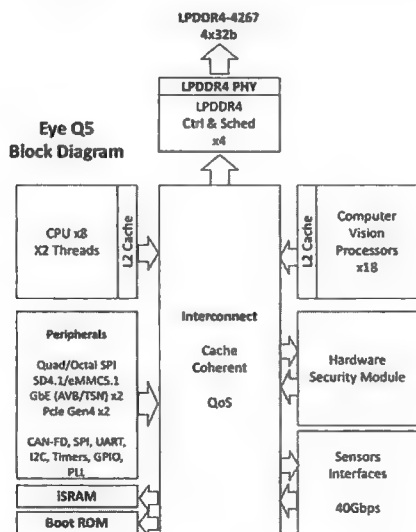


图 9-18 MobilEye EyeQ5 结构示意图

此外, Nervana 一直在努力将机器学习功能全力引入到芯片之中, 是人工智能 ASIC 芯片供应商。得到 Intel 的支持后, Nervana 正计划推出其针对深度学习算法的定制芯片 Nervana Engine。据 Nervana 相关人员表示, 相比 GPU, Nervana Engine 在训练方面可以提升 10 倍性能。借助 Nervana Engine 芯片在深度学习训练方面优于传统 GPU 的能耗和性能优势, Intel 也相继推出了一系列适应深度神经网络的特殊处理器。

5. 其他芯片解决方案

1) 谷歌 TPU 芯片

谷歌公布了 AlphaGo 战胜李世石的“秘密武器”就是芯片“TPU”(张量处理单元, Tensor Processing Unit), 它使得机器学习类深度神经网络模型在每瓦特性上优于传统硬件。TPU 在 Google 2016 I/O 上首次被提及, 然而谷歌早在 2013 年就开始秘密研发 TPU, 并且在 2014 年就已应用于谷歌的数据中心。TPU 专为谷歌 TensorFlow 等机器学习应用打造, 能够降低运算精度, 在相同时间内处理更复杂、更强大的机器学习模型并将其更快投入使用。其性能把人工智能技术往前推进了差不多 7 年, 相当于摩尔定律 3 代的时间。

相比 GPU 更适合训练, TPU 更适合做训练后的分析决策。这一点在谷歌的官方声明里也得到了印证: TPU 只在特定机器学习应用中起辅助使用, 公司将继续使用其他厂商制造的 CPU 和 GPU。因此, TPU 再好, 也仅适用于谷歌, 而且还是用于辅助 CPU 和 GPU。

2) 后起之秀概率芯片

2016 年 4 月 16 日, MIT Technology Review 报道, DARPA 投资了一款由美国 Singular Computing 公司开发的“S1”概率芯片, 如图 9-19 所示。模拟测试中, 使用 S1 追踪视频里的移动物体, 每帧处理速度比传统处理器快了近 100 倍, 而能耗还不到传统处理器的 2%。专用概率芯片可以发挥概率算法简单并行的特点, 极大地提高系统性能。其优点包括算法逻辑异常简单, 不需要复杂的数据结构, 不需要数值代数计算; 计算精度可以通过模拟不同数目的随机行走自如控制; 不同的随机行走相互独立, 可以大规模并行模拟; 模拟过程中, 不需要全局信息, 只需要网络的局部信息。

早在 2008 年 MIT Technology Review “十大科技突破”预测中, 概率芯片就榜上有名。通过牺牲微小的计算精度换取能耗明显降低, 概率芯片在历来追求精准的芯片领域独树一帜, 但正因如此, 概率芯片很可能后来居上。

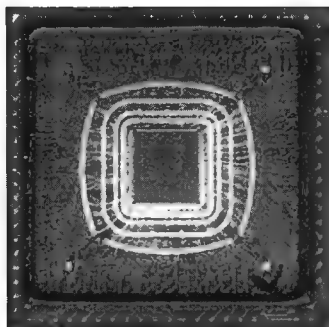


图 9-19 “S1” 概率芯片示意图（来源：MIT Technology Review）

3) 中国的芯片方案

寒武纪：中国的智能芯片

寒武纪是国际上首个深度学习专用处理器芯片。2012 年，中科院计算所和法国 Inria 等机构共同提出了国际上首个人工神经网络硬件的基准测试集 benchNN。此后，中科院计算所和法国 Inria 的研究人员共同推出了一系列不同结构的 DianNao 神经网络硬件加速器结构。当前，寒武纪系列已包含四种处理器结构：DianNao（面向多种人工神经网络的原型处理器结构）^[15]、DaDianNao（面向大规模人工神经网络）^[16]和 PuDianNao（面向多种机器学习算法）^[17]，面向卷积神经网络的 ShiDianNao^[18]。寒武纪进入产业化运营，其主要方向是高性能服务器芯片、高效终端芯片和服务机器人芯片，寒武纪芯片预计于 2018 年左右上市。

中星微

2016 年 6 月，中星微率先推出了中国首款嵌入式神经网络处理器（NPU）芯片“星光智能一号”，这也是全球首枚具备深度学习人工智能的嵌入式视频采集压缩编码系统级芯片，并已于 3 月 6 日实现了量产。该芯片采用了“数据驱动”并行计算的架构，单颗 NPU（28nm）能耗仅为 400mW，极大地提升了计算能力与功耗的比例，可以广泛应用于智能驾驶辅助、无人机、机器人等嵌入式机器视觉领域。

9.3.3 计算平台体系结构设计探索

本节，我们尝试对以下问题形成一些初步认识。

（1）各种计算单位最适合什么样的工作负载。

(2) 能否使用移动处理器执行无人驾驶计算任务。

(3) 如何设计一个高效的无人驾驶计算平台^[19]。

1. 计算单元与计算负载的匹配

我们试图了解哪些计算单元最适合执行卷积和特征提取类应用,这是无人驾驶场景中最计算密集型的工作负载。我们在现有的 ARM SOC 上完成了实验验证,此 ARM SOC 由一个四核 CPU、GPU、DSP 组成,详细硬件数据可见。为了研究各种异构硬件的能耗与性能行为,我们分别在 CPU、GPU、DSP 实现并优化了特征提取和卷积这两类计算负载,同时测量了芯片级能耗。

首先,我们分别在 CPU、GPU 和 DSP 实现了卷积应用,这是在对象识别和目标跟踪任务中最常用、计算也最密集的阶段。当在 CPU 上运行时,每次卷积大约需要 8 毫秒来完成,能耗为 20 MJ;在 DSP 上运行时,每次卷积需要 5 毫秒来完成,能耗为 7.5 MJ;在 GPU 运行时,每次卷积只需要 2 毫秒来完成,能耗也仅需 4.5 MJ。这表明,无论是性能还是能耗表现,GPU 是执行卷积任务最有效的计算单元,如图 9-20 所示。

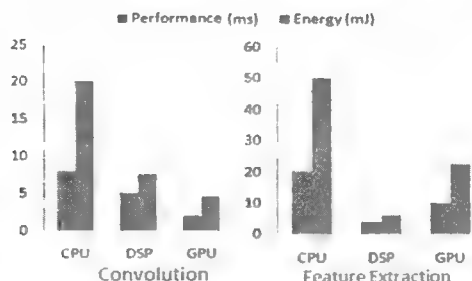


图 9-20 卷积应用和特征提取在 CPU、GPU 和 DSP 上的性能表现

接下来,我们实现了分别在 CPU、GPU 和 DSP 特征提取应用。特征提取为无人驾驶的定位产生特征点,这是定位阶段计算量最大的工作负载:在 CPU 上运行时,每个特征提取的任务大约需要 20 毫秒来完成,耗能 50 MJ;在 GPU 上运行时,每个特征提取的任务需要 10 毫秒来完成,耗能 22.5 MJ;在 DSP 中运行时,每个特征提取的任务仅需要 4 毫秒,仅消耗 6 MJ。这些结果表明,从性能和能耗的角度出发,DSP 是特征提取最有效的执行计算单元。我们并没有对无人驾驶中的其他任务,如定位、规划、避障等进行上述分析,这是因为对 GPU 和 DSP 这类专注于并行的硬件而言,上述任务侧重于控制逻辑因此得不到高效执行。

2. 移动处理器上的无人驾驶

我们尝试了解无人驾驶系统在上述 ARM 移动 SoC 上的执行情况，并探索支持无人驾驶的最低硬件平台配置。图 9-21 所示为一个面向基于视觉的无人驾驶的移动 SoC 系统。在这个移动 SoC 实现中，我们利用 DSP 处理传感器数据，如特征提取和光流；我们使用 GPU 完成深度学习任务，如目标识别；采用两个 CPU 线程完成定位任务以实现车辆实时定位；我们使用一个 CPU 线程实现实时路径规划；使用另一个 CPU 线程进行避障操作。如果 CPU 尚未被全占有，则多个 CPU 线程可以在同一个 CPU 核心上运行。

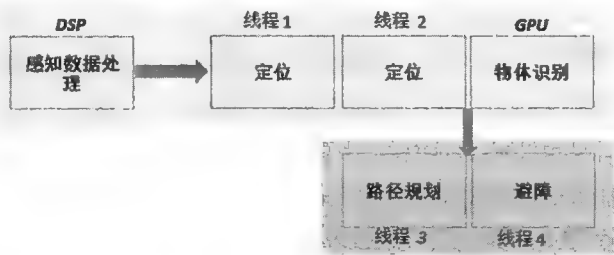


图 9-21 无人驾驶的 Mobile SoC 设计结构示意图

令人惊讶的是，实验数据证明无人驾驶系统在 ARM SOC 上运行的性能并不差。定位流水线每秒可处理 25 帧图像，图像生成速度为每秒 30 帧图像，这说明产生的图像大部分可以得到及时处理，不会产生大规模的丢帧。深度学习流水线每秒能够执行 2~3 个目标识别任务。规划和控制流水线目标是在 6ms 内完成路径规划。当使用 ARM 移动端 SoC 进行无人驾驶时，车辆能够以 5 英里/小时的速度行驶，并且不损失任何定位信息。同时，整个 SoC 平均功耗为 11W。移动 SoC 的硬件资源有限，能够支持有限范围内的无人驾驶系统确实是非常令人惊喜的发现。这说明如果增加更多的计算资源，硬件平台就能够处理更多的数据，并支持车辆以更快的速度行驶，最终满足产品级无人驾驶系统的需要。

3. 计算平台的设计

ARM 移动端 SoC 之所以能提供这样的性能，是因为我们充分利用了硬件系统的异构计算资源，为每一个不同的无人驾驶子任务匹配最适合的计算单元，以达到最优化的性能和能源效率。然而，这样的设计思路仍然存在一个缺点：我们不可能为所有的子任务找到适配的计算单元。例如，目标跟踪、变更车道预测、交叉道路交通流量预测等逻辑与计算都较为密集的子任务。此外，我们希望成熟的无人驾驶系统能够上传原始传感器数据，并在云端完成数据处理。然而，传感器数据量非常巨大，无人驾驶系统应设计为有能力采用

一切可用的网络带宽，在短时间内完成数据上传。

上述物体跟踪、数据上传等子任务在整个无人驾驶周期内并不需要一直运行。例如，只有物体识别会触发物体跟踪，只有物体跟踪才能触发流量预测。数据的批量上传可提高系统吞吐量并减少带宽的使用，因此数据上传只会在某一段时间内存在。如果为每一个这样暂存的子任务设计其专有的 ASIC 芯片，在制造成本和人力成本上都得不偿失。相反，FPGA 是这些短周期子任务的完美适配。在系统中我们仅需要一个 FPGA 芯片，通过部分重构技术，这些子任务可以分时共享 FPGA。部分重构技术已被证明可在小于几毫秒的时间内完成 FPGA 内核的重构，实时实现分时共享。

在图 9-22 中，我们提出了面向驾驶的无人计算堆栈。在计算平台层，我们提出了一个新的 SoC 架构。在此 SoC 架构中，有一个 I/O 子系统与前端传感器交互；由 DSP 负责图像预处理流以进行特征提取；由 GPU 进行目标识别和其他深度学习任务；由一个多核 CPU 完成规划、控制和互动的子任务；由 FPGA 进行动态重构以分时共享的方式完成传感器数据压缩上传、物体跟踪和流量预测等工作。计算部件和 I/O 部件之间通过共享内存进行数据通信。在 SoC 硬件平台上有一个动态系统，通过 OpenCL 把不同的工作负载分配到异构的计算单元上执行，并由实时的执行引擎动态地完成任务调度。在动态系统之上部署的是机器人操作系统（ROS）。ROS 是一个分布式操作系统，其中包含多个 ROS 节点，每个节点上执行一个无人驾驶子任务，节点之间相互通信进行多任务协调。

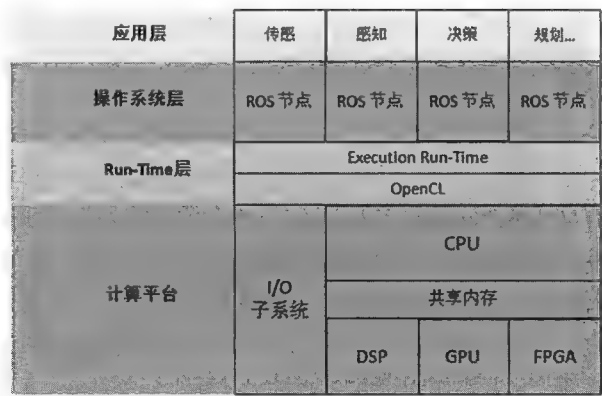


图 9-22 无人驾驶计算栈结构示意图

4. 讨论与结论

我们已经实现了上述无人驾驶计算堆栈，相比已有的无人驾驶设计平台，我们的设计

有以下优点。

- (1) 模块化：如果需要更多的功能可添加更多的 ROS 节点。
- (2) 安全性：ROS 节点提供一个良好的隔离机制，防止节点相互影响。
- (3) 高度动态化：动态系统层可根据需要完成调度以实现大吞吐量、低延迟或低能耗。
- (4) 高性能：异构的体系结构可以保证每个专用计算单元为适配的子任务提供最高性能。
- (5) 节能性：专用的计算单位为每个子任务提供了最高效的运算方式，例如 DSP 模块是面向特征提取子任务中最具能耗有效性的执行单元。

现有的面向 4 级无人驾驶的计算平台解决方案功耗往往耗资数万美元，不仅功耗高达数千瓦，在运行时也将产生大量的热量，机器发热严重，严重威胁系统运行的可靠性。这些功耗、散热和制造成本上存在的问题使得无人驾驶技术难以服务于一般公众。在本节，我们提出并实现了一个模块化的、更安全的、更高性能、能耗更有效的无人驾驶计算架构和软件堆栈。我们在 ARM 移动端 SoC 上运行原型系统平均功耗为 11 W，能驱动车辆以每小时 5 英里的速度实现无人驾驶。这说明如果增加更多的计算资源，系统能够处理更多的数据，最终将满足产品级无人驾驶系统的需要。

9.4 控制平台

控制平台是无人车的核心部件，控制着车辆的各种控制系统，包括汽车防抱死制动系统（ABS）、汽车驱动防滑转系统（ASR）、汽车电子稳定程序（ESP）、电子感应制动控制系统（SBC）、电子制动力分配（EBD）、辅助制动系统（BAS）、安全气囊（SRS）和汽车雷达防碰撞系统、电控自动变速器（EAT）、无级变速器（CVT）、巡航控制系统（CCS）、电子控制悬架（ECS）、电控动力转向系统（EPS）等。控制平台主要包括电子控制单元（ECU）与通信总线两大部分：ECU 主要实现控制算法，通信总线主要实现 ECU 及机械部件间的通信功能。本节我们将详细介绍控制平台。

9.4.1 电子控制单元

电子控制单元（ECU，Electronic Control Unit），俗称“车载电脑”，是汽车专用微机

控制器,也叫汽车专用电脑。发动机工作时,ECU 采集各传感器的信号进行运算,并将运算结果转变为控制信号,控制被控对象的工作。固有程序在发动机工作时,不断地与采集来的各传感器的信号进行比较和计算,再利用比较和计算后的结果完成对发动机的点火、怠速、废气再循环等多项参数的控制。它还有故障自诊断和保护功能。存储器也会不停地记录行驶中的数据,成为 ECU 的学习程序,为适应驾驶习惯提供最佳的控制状态,这叫自适应程序。在高级轿车上,有不止一只 ECU 如防抱死制动系统、四轮驱动系统、电控自动变速器、主动悬架系统、安全气囊系统、多向可调电控座椅等都配置有各自的 ECU。随着轿车电子化、自动化的提高,ECU 将日益增多,线路会日益复杂。宝马、奔驰和奥迪三大车厂各系列高阶车款皆已包含超过一百个 ECU。ECU 的电压工作范围一般在 6.5V~16V (内部关键处有稳压装置)、工作电流在 0.015A~0.1A、工作温度在零下 40~80 度,能承受 1000Hz 以下的振动,损坏率非常小。

ECU 从用途上讲是汽车专用微机控制器,也叫汽车专用单片机。它和普通的单片机一样,由微处理器 (CPU)、存储器 (ROM、RAM)、输入/输出接口 (I/O)、模数转换器 (A/D) 及整形、驱动等大规模集成电路组成。存储器 ROM 中存储的是一套固定的程序,该程序是以精确计算和大量实验取得的数据为基础。固有程序在发动机工作时,不断地与采集来的各传感器的信号进行比较和计算,然后输出指令,以控制发动机的点火、空燃比、怠速、废气再循环等多项参数的设置,判断是否需要改变喷油量、点火时间及气门开度的大小等。

详细来说,当发动机起动时,电控单元进入工作状态,某些程序从 ROM 中取出进入 CPU,这些程序专用于控制点火时刻、控制汽油喷射、控制怠速等。执行程序中所需的发动机信息来自各个传感器。这些传感器信号一经采集首先进入输入回路接受处理,如果是模拟信号,则需先经过 A/D 转换器转换成数字信号。大多数传感器信息将先暂存在 RAM 内,然后根据程序处理顺序从 RAM 送至 CPU。接下来是将存储器 ROM 中的参考数据引入 CPU,与传感器输入数据进行比较。CPU 在完成对这些数据的比较运算后,做出决定并发出指令信号,经 I/O 接口进行放大,必要的信号还经 D/A 转换器变成模拟信号,最后经输出回路控制执行器动作。

随着轿车电子化、自动化的提高,ECU 将日益增多,目前高端汽车在总计 100 多个 ECU 系统中包含多达 200 个微处理器。这数百个 ECU,在汽车内部组成了一个区域网。一个 ECU 发出的数据包,所有的节点都会接收到,但只有承担该数据包任务的节点才会去执行命令。以刹车灯为例,当监控刹车踏板的 ECU 监测到踏板行程有变动时,会通知

监测尾灯的 ECU。此时，该 ECU 控制尾灯，并将其通电点亮。这个简单的操作其实背后有至少两个 ECU 的配合。要让所有这些 ECU 之间相互配合，就需要采用一种称为多路复用的通信网络协议进行信息传递，控制器区域网（CAN，Controlllers Area Network）总线是其中之一。

借助 CAN 协议，汽车内部的数百个 ECU 可以组建一个区域网，有效地解决线路信息传递所带来的复杂化问题。通用、沃尔沃、特斯拉等车型支持远程控制，其原理就是手机发出的指令先到达伺服器，然后被转发到车载通信模块。车载通信模块接收到指令后，再通过 CAN 总线将指令传达到各个 ECU。

为了弥补 CAN 协议在某些方面的不足，汽车工业还研发出了很多其他协议，比如 LIN 协议。相比 CAN，LIN 的带宽更小、承载的数据量更少，同时成本也更低，适合应用在一些简单的 ECU 中，比如车窗升降等。随着技术的进步，汽车内部的数据量暴增，尤其是大屏幕的普及和流媒体技术的介入，让 CAN 总线在某些时候“力不从心”，已无法胜任工作。于是，更高级的通信协议问世了，比如 MOST、FlexRay、以太网等。这些协议标准，拥有更大的带宽与更强的稳定性。其中，MOST 是一种高速多媒体传输接口，专门为汽车内部的一些高码率音频、视频提供传输。FlexRay 也是一种高速协议，但不仅限于多媒体传输。在自动驾驶的奥迪 A7 中，位于后备箱的车载 CPU（奥迪称之为 zFAS）模组，就是依靠 FlexRay 协议读取前置摄像头捕捉的数据。

EUC 的主要生产厂商括有博世（BOSCH）、德尔福（DELPHI）、马瑞利（MARELLI）、日立（Hitachi）、大陆（Continental）、日本电装（DENSO）等。主要产品包括博世的 M7、M7.9.7、M7.9.7.1、ME7、ME7.9.7、ME7.8.8、EDC16、054K0；德尔福的 MT20U2、MT20U、MT20、MR140、MT80、ITMS-6F；日立/电装的 69J0、69EB、77J0、3601015A28K；西门子的 SIM2K-34、SIM2k-51.4、SIM2k-D51、SIMK43 等系列。

CPU 是 ECU 中的核心部分，它具有运算与控制的功能。在发动机运行时，它采集各传感器的信号进行运算，并将运算的结果转变为控制信号，控制被控对象的工作。它还实行对存储器（ROM、RAM）、输入/输出接口（I/O）和其他外部电路的控制。Power Train ECU 采用的 CPU 基本来自于 Infineon、ST 和 Freescale。BOSCH 的 16 位 ECU M(E)7 系列早期主要使用 Infineon C167 内核的 CPU。之后 ST 为 BOSCH 定制了 ST10 系列 CPU，价格上更有优势，因此 BOSCH 后期的 16 位 ECU 基本上都采用 ST10 系列 CPU。BOSCH 的 32 位 ECU ME9 系列主要使用 Freescale 的 PowerPC 内核的 CPU MPC55 系列，其中 ME9 在美国市场上销售的 MED17 系列则使用基于 Infineon Tricore 内核的 CPU TC17xx。

MED17 系列 ECU 有好多分枝, 分别使用不同型号的 TC17xx CPU。MEDC18 系列依然沿用 PowerPC 路线, 选择了 ST 和 Freescale 两家供应商, 使用了 Freescale 的 XPC56 系列 CPU 及 ST 的 SPC56 系列 CPU。车身 ECU 的选择更多, Infineon、ST、Freescale、NEC 和瑞萨电子都提供相关 CPU 的支持。

9.4.2 通信总线

随着汽车各系统的控制逐步向自动化和智能化转变, 汽车电气系统变得日益复杂。为了满足各电子系统的实时性要求, 我们须对汽车数据, 如发动机转速、车轮转速、节气门踏板位置等信息实行共享, 因而我们需要汽车通信总线。如图 9-23 所示, 目前, 车用总线技术被美国汽车工程师协会 SAE 下属的汽车网络委员会按照协议特性分为 A、B、C、D 四类。

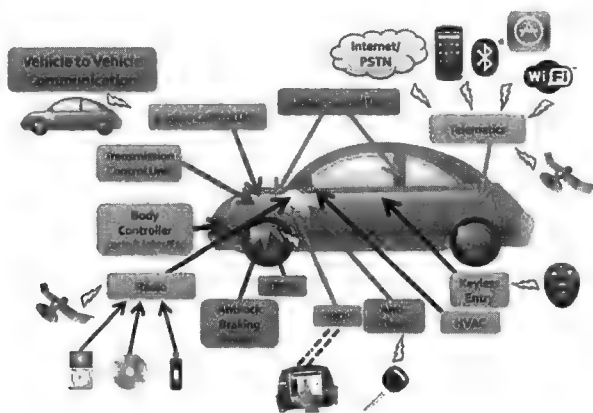


图 9-23 车用通信总线分布示意图

A 类总线面向传感器或执行器管理的低速网络, 它的位传输速率通常小于 20Kb/s。A 类总线以 LIN (Local Interconnect Network, 本地互联网) 规范为代表, 是由摩托罗拉与奥迪等企业联手推出的一种新型低成本的开放式串行通信协议, 主要用于车内分布式电控系统, 尤其是面向智能传感器或执行器的数字化通信场合。

B 类总线面向独立控制模块间信息共享的中速网络, 位速一般在 10~125 Kb/s。B 类总线以 CAN (Controller Area Network, 控制器局域网) 为代表。CAN 网络最初是 BOSCH 公司为欧洲汽车市场开发的, 只用于汽车内部测量和执行部件间的数据通信, 逐渐地发展完善技术和功能, 1993 年, ISO 正式颁布了道路交通运输工具——数字信息交换——高

速通信控制器局域网 (CAN) 国际标准 (ISO11898-1), 近几年低速容错 CAN 的标准 ISO 11519-2 也开始在欧洲的一些车型中得到广泛应用。

C 类总线面向闭环实时控制的多路传输高速网络, 位速率多在 125Kb/s ~ 1Mb/s。C 类总线主要用于车上动力系统中对通信的实时性要求比较高的场合, 主要服务于动力传递系统。汽车厂商大多使用“高速 CAN”作为 C 类总线, 它实际上就是 ISO 11898-1 中位速率高于 125Kb/s 的那部分标准。

D 类总线面向多媒体设备、高速数据流传输的高性能网络, 位速率一般在 2Mb/s 以上, 主要用于 CD 等播放机和液晶显示设备。D 类总线带宽范畴相当大, 用到的传输介质也有好几种, 其又被分为低速 (IDB-C 为代表)、高速 (IDB-M 为代表) 和无线 (Bluetooth 为代表) 三大范畴。

下面我们主要了解局部互联协议 LIN、控制器局域网 CAN, 以及高速容错网络协议 FlexRay。

1. 局部互联协议 LIN

LIN 是面向汽车低端分布式应用的低成本、低速串行通信总线。它的目标是为现有汽车网络提供辅助功能, 在不需要 CAN 总线的带宽和多功能的场合使用, 降低成本。LIN 相对于 CAN 的成本节省来自三方面: 采用单线传输、硅片中硬件或软件的低成本实现及无须在从属节点中使用石英或陶瓷谐振器。这些优点是以较低的带宽和受局限的单宿主总线访问方法为代价的。LIN 采用单个主控制器多个从设备的模式, 在主从设备之间只需要一根电压为 12V 的信号线。这种主要面向“传感器/执行器控制”的低速网络, 其最高传输速率可达 20Kb/s, 应用于电动门窗、座椅调节、灯光照明等控制系统。典型的 LIN 网络的节点数可以达到 12 个。以门窗控制为例, 在车门上有门锁、车窗玻璃开关、车窗升降电机、操作按钮等, 只需要 1 个 LIN 网络就可以把它们连为一体。通过 CAN 网关, LIN 网络还可以和汽车的其他系统进行信息交换, 实现更丰富的功能, 如图 9-24 所示。

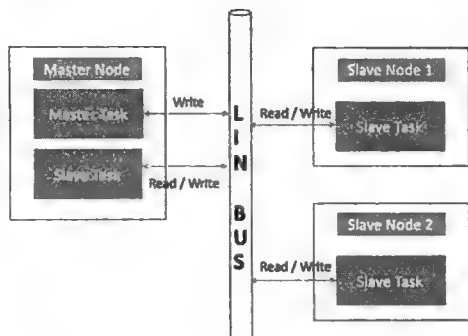


图 9-24 LIN 总线信息交换示意图

LIN 包含一个宿主节点 (Master) 和一个或多个从属节点 (Slave)。所有节点都包含一个被分解为发送和接收任务的从属通信任务,而宿主节点还包含一个附加的宿主发送任务。在实时 LIN 中,通信总是由宿主任务发起的。除了宿主节点的命名外,LIN 网络中的节点不使用有关系统设置的任何信息。我们可以在不要求其他从属节点改变硬件和软件的情况下向 LIN 中增加节点。宿主节点发送一个包含同步中断、同步字节和消息识别码的消息报头,从属任务在收到和过滤识别码后被激活并开始消息响应的传输。响应包含两个、四个或八个数据字节和一个检查和 (checksum) 字节。报头和响应部分组成一个消息帧。LIN 总线上的所有通信都由主机节点中的主机任务发起,主机任务根据进度表确定当前的通信内容,发送相应的帧头,并为报文帧分配帧通道。总线上的从机节点接收帧头之后,通过解读标识符确定自己是否应该对当前通信做出响应、做出何种响应。基于这种报文滤波的方式,LIN 可实现多种数据传输模式,且一个报文帧可以同时被多个节点接收利用。

2. 控制器局域网 CAN

在当前的汽车总线网络市场上,占据主导地位的是 CAN 总线。CAN 总线是德国博世公司在 20 世纪 80 年代初为了解决现代汽车中众多的控制与测试仪器之间的数据交换问题而开发的一种串行数据通信协议。它的短帧数据结构、非破坏性总线性仲裁技术及灵活的通信方式适应了汽车的实时性和可靠性要求。CAN 总线分为高速和低速两种,高速 CAN 最高速度为 1Mbit/s (C 类总线),低速 CAN 为 250Kbit/s (B 类总线),如图 9-25 所示。

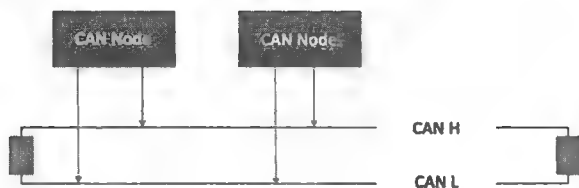


图 9-25 CAN 总线结构图

CAN 总线一般为线型结构，所有节点并联在总线上。当一个节点损坏时，其他节点依然能正常工作。但当总线一处出现短路时，整个总线便无法工作。CAN 总线采用 CSMA/CA（Carrier Sense Multiple Access with Collision Avoidance）机制。各节点会一直监听总线，发现总线空闲时便开始发送数据。当多个节点同时发送数据时，会通过一套仲裁机制竞争总线。每个节点会先发送数据的 ID，ID 越小表示优先级越大，优先级大的会自动覆盖小的 ID。当节点发现自己发送的 ID 被覆盖掉时，就知道有比他优先级更高的消息正在被发送，便自动停止发送。优先级最高的消息获得总线使用权，开始发送数据。当高优先级的数据包发送完后，各节点便又尝试竞争总线。如此反复下去。这样能最大程度地利用总线。弊端是会有时效延迟，优先级越低的数据包，可能需要等待的时间越长。从这一点上来讲，CAN 总线不是一种实时总线。当 CAN 总线有节点发现当前发送的数据有误时，会发送错误帧告知总线上的所有节点。发送错误数据的节点会重发。每个节点都有一个错误计数器。当一个节点总是发送或接收错误超过一定次数时，会自动退出总线。

3. 高速容错网络协议 FlexRay

FlexRay 总线数据收发采取时间触发和事件触发的方式。利用时间触发通信时，网络中的各个节点预先知道彼此将要进行通信的时间，接收器提前知道报文到达的时间，报文在总线上的时间可以预测出来。即便行车环境恶劣多变，干扰了系统传输，FlexRay 协议也可以确保将信息延迟和抖动降至最低，尽可能保持传输的同步与可预测。这对需要持续及高速性能的应用（如线控刹车、线控转向等）来说是非常重要的。

FlexRay 总线用的是 TDMA（Time Division Multiple Access）和 FTDMA（Flexible Time Division Multiple Access）两种周期通信方法。FlexRay 将一个通信周期分为静态部分、动态部分和网络空闲时间。静态部分使用 TDMA 方法，每个节点会均匀分配时间片，每个节点只有在属于自己的时间片里才能发送消息，即使某个节点当前无消息可发，该时间片依然会保留（也就造成了一定的总线资源浪费）。在动态部分使用 FTDMA 方法会轮流回

询每个节点有没有消息要发，有就发，没有就跳过。静态部分用于发送需要经常性发送的重要性高的数据，动态部分用于发送使用频率不确定、相对不重要的数据。当 FlexRay 总线通信过程中出现数据错误时，该周期里接收到的所有数据都会被丢弃掉，但没有重发机制。所有节点会继续进行下一个周期的通信。FlexRay 同样也有错误计数器，当一个节点发送接收错误过多时会被踢出总线。

FlexRay 具有高速、可靠及安全的特点。FlexRay 在物理上通过两条分开的总线通信，每一条的数据速率是 10MBit/s。FlexRay 还能提供很多网络所不具有的可靠性特点，尤其是 FlexRay 具备的冗余通信能力可实现通过硬件完全复制网络配置，并进行进度监测。FlexRay 同时提供灵活的配置，可支持各种拓扑，如总线、星型和混合拓扑。FlexRay 本身不能确保系统安全，但它具备大量功能，可以支持以安全为导向的系统（如线控系统）的设计。

宝马公司在 2007 款 X5 系列车型的电子控制减震器系统中首次应用了 FlexRay 技术，如图 9-26 所示。此款车采用基于飞思卡尔的微控制器和恩智浦的收发器，可以监视有关车辆速度、纵向和横向加速度、方向盘角度、车身和轮胎加速度及行驶高度的数据，实现了更好的乘坐舒适性及驾驶时的安全性和高速响应性，此外还将施加给轮胎的负荷变动及底盘的振动均减至最小。

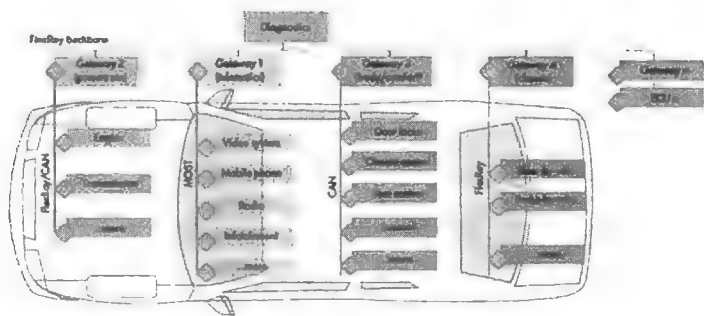


图 9-26 FlexRay 总线分布图

9.5 结论

如果说算法是无人驾驶的灵魂，那么硬件平台就是无人驾驶的肉体。一个没有肉体的灵魂也只是孤魂野鬼而已，再高大上的算法也需要在硬件平台上才有实用价值。硬件平台

的设计直接决定了无人驾驶对环境的感知能力、计算性能与能耗、鲁棒性、安全性等。无人驾驶的硬件平台又分为传感器平台、计算平台，以及控制平台三部分。本章详细介绍了这三种平台及现有的解决方案。希望本章对无人驾驶从业者及爱好者选择硬件有帮助。

9.6 参考资料

- [1] Velodyne Product List, <http://velodynelidar.com/products.html>.
- [2] Ibeo, <https://www.ibeo-as.com/>.
- [3] Quanergy solid State S3, <http://quanergy.com/s3/>.
- [4] Velodyne VLP-16 PUC, <http://velodynelidar.com/vlp-16.html>.
- [5] 北醒光子激光雷达产品列表, <http://benewake.com/Product.html>.
- [6] 镭神智能激光雷达产品列表, <http://www.leishenlidar.com/product/leida/index.html>.
- [7] 速腾聚创多线激光雷达, <http://www.robosense.cn/web/rslidar/cn>.
- [8] 关于发布《微功率(短距离)无线电设备的技术要求》的通知, <http://www.miit.gov.cn/n1146295/n1146592/n3917132/n4062354/n4062391/n4062397/n4062399/c4148119/content.html>.
- [9] 舜宇车载摄像头解决方案, <http://www.sunnyoptical.com/pro/001016/index.html>.
- [10] 国家发展改革委交通运输部关于印发《推进“互联网+”便捷交通促进智能交通发展的实施方案》的通知, http://www.sdpc.gov.cn/gzdt/201608/t20160805_814075.html.
- [11] NVIDIA PX2, <http://www.nvidia.cn/object/drive-px-cn.html>.
- [12] 德州仪器 TDA2x SoC, <http://processors.wiki.ti.com/index.php/TDA2x>.
- [13] CEVA XM4 - Intelligent Vision Processor, <http://www.ceva-dsp.com/CEVA-XM4>.
- [14] Cyclone V SoCs, <https://www.altera.com.cn/products/soc/portfolio/cyclone-v-soc/overview.html>.
- [15] Tianshi Chen, Zidong Du, Ninghui Sun, and et al.. DianNao: a small footprint

- high-throughput accelerator for ubiquitous machine-learning. In *Proceedings of the 19th international conference on Architectural support for programming languages and operating systems - ASPLOS'14*, pages 269-284, Salt Lake City, UT, USA, 2014.
- [16] Yunji Chen, Tao Luo, Shaoli Liu, and et al.. DaDianNao: A Machine- Learning Supercomputer. In *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture -MICRO-47*, 2014.
- [17] Daofu Liu, Tianshi Chen, Shaoli Liu, Jinhong Zhou, Shengyuan Zhou, Olivier Temam, Xiaobing Feng, Xuehai Zhou, and Yunji Chen. PuDianNao: A Machine Learning Accelerator. In *Proceedings of the 20th international conference on Architectural support for programming languages and operating systems*, pages 369-381, 2015.
- [18] Zidong Du, Robert Fasthuber, Tianshi Chen, Paolo Ienne, Ling Li, Tao Luo, Xiaobing Feng, Yunji Chen, and Olivier Temam. Shidiannao: Shifting vision processing closer to the sensor. In *Proceedings of the 42nd ACM/IEEE International Symposium on Computer Architecture (ISCA'15)*. ACM, 2015.
- [19] Shaoshan Liu, Jie Tang, Zhe Zhang, Jean-Luc Gaudiot: CAAD, Computer Architecture for Autonomous Driving, *IEEE Computer*, <https://arxiv.org/abs/1702.01894>.

10

无人驾驶系统安全

本章主要介绍无人驾驶安全。对无人驾驶系统来说，安全性是至关重要的，但目前针对无人车攻击的方法有许多。如何防御这些攻击以保证无人车的安全是个重要的课题。本节将详细介绍针对无人车传感器、操作系统、控制系统、车联网的攻击手段及防御方法。

10.1 针对无人驾驶的安全威胁

对于无人驾驶系统来说，安全性是至关重要的。任何无人车如果达不到安全要求就上路是极其危险的。目前，针对无人车攻击的方法五花八门，渗透到无人驾驶系统的每个层次，包括传感器、操作系统、控制系统、车联网通信系统等。首先，针对传感器的攻击不需要进入无人驾驶系统内部，这种外部攻击法技术门槛相当低，既简单又直接。第二，如果进入了无人驾驶操作系统，黑客可以造成系统崩溃导致停车，也可以窃取车辆敏感信息。第三，如果进入了无人驾驶控制系统，黑客可以直接操控机械部件，劫持无人车去伤人，是极其危险的。第四，车联网连接不同的无人车，以及中央云平台系统，劫持车联网通信系统也可以造成无人车间的沟通混乱。^{[1][2][3]}

10.2 无人驾驶传感器的安全

由于传感器处于整个无人驾驶计算的最前端，最直接攻击无人车的方法就是攻击传感器。

这种外部攻击法并不需要入侵到无人驾驶系统内部,使得入侵的技术门槛相当低。^{[4] [5]}正是因为入侵的门槛低,我们需要在传感器端做大量的工作来保证其安全。如图 10-1 所示,对各种传感器,我们都可以轻易地攻击与误导。在之前的章节中我们提到可以使用惯性传感器 IMU 辅助无人驾驶定位,但是 IMU 对磁场很敏感,如果我们使用强磁场干扰 IMU,就有可能影响 IMU 的测量。对于 GPS,如果在无人车附近设置大功率假 GPS 信号,就可以覆盖原来的真 GPS 信号,从而误导无人车的定位。通过两种简单攻击方法的结合, GPS 与 IMU 的定位系统会被轻易攻破。除了 GPS 与 IMU 外,通常我们也可以使用轮测距技术辅助无人车定位。轮测距是通过测量轮子的转速乘以轮子的周长进行测距,如果黑客破坏了轮子,这个定位辅助技术也会受影响。

激光雷达是目前无人驾驶最主要的传感器,而无人车也依赖于激光雷达数据与高精地图的匹配进行定位。但是激光雷达也可以被轻易干扰。首先,激光雷达是通过测量激光反射时间来测量深度的。如果我们在无人车周围放置强反光物,比如镜子,那么激光雷达的测量就会被干扰,返回错误信息。除此之外,如果黑客使用激光照射激光雷达,激光雷达的测量也会受干扰,会分不清哪些是自身发出的信号,哪些是外部激光的信号。另外,无人车会不断地下载更新的高精地图,如果黑客把下载的地图掉包,也会造成定位失效。

计算机视觉可以辅助无人车完成许多感知的任务,比如交通灯识别、行人识别、车辆行驶轨迹跟踪等。在交通灯识别的场景中,无人车上的摄像机如果检测到红灯,那么无人车就会停下来。如果检测到行人,无人车也会停下来以免发生意外。黑客可以轻易地在路上放置假的红绿灯及假的行人,迫使无人车停车并对其进行攻击。^[6]

那么,既然每个传感器都可以被轻易的攻击,如何保证无人车的安全呢?对此,我们需要使用多传感器融合的技术,使用多个传感器互相纠正。攻击单个传感器很容易,但是如果同时攻击所有的传感器难度相当大。当无人车发现不同传感器的数据相互间不一致时,就知道自己有可能正在被攻击。简单的例子是如果无人车检查到交通灯,但是高精地图在此处并未标注有交通灯,那么很可能是被攻击了。又例如, GPS 系统与 LiDAR 系统定位的位置极不一致,那么无人车也很可能是被攻击了。



图 10-1 针对传感器的攻击示意图

10.3 无人驾驶操作系统的安全

针对传感器的攻击是外部攻击，并不需要进入无人驾驶系统。另一种攻击方式是入侵到无人驾驶操作系统，劫持其中一个节点并对其进行攻击。目前的无人驾驶操作系统基本是基于 ROS 的框架实现的，ROS 本身安全性有一定的问题，简单总结有以下攻击方法。^[7] 第一，其中一个 ROS 的节点被劫持，然后不断地进行分配内存，导致其系统内存消耗殆尽，造成系统 OOM 开始关闭不同的 ROS 节点进程，使整个无人驾驶系统崩溃。第二，ROS 的 topic 或者 service 被劫持，导致 ROS 节点之间传递的信息被伪造，从而导致无人驾驶系统的异常行为。造成第一个问题的原因是 ROS Node 本身是一个进程，可以无节制地分配资源导致崩溃，另一个原因是 ROS Node 可以访问磁盘及网络资源，并无很好的隔离机制。为了解决这个问题，我们可以使用 Linux 容器技术（LXC）管理每一个 ROS 节点的进程。简单来说，LXC 提供轻量级的虚拟化，以便隔离进程和资源，而且不需要提供指令解释机制及全虚拟化等其他复杂功能，相当于 C++ 中的 NameSpace。LXC 有效地将由单个操作系统管理的资源划分到孤立的群组中，以更好地在孤立的群组之间平衡有冲突的资源使用需求。对于无人驾驶场景来说，LXC 最大的好处是性能损耗小。我们测试发现，在运行时 LXC 只造成了 5% 左右的 CPU 损耗。除了资源限制外，LXC 也提供了沙盒支持，使得系统可以限制 ROS 节点进程的权限。为了避免可能有危险性的 ROS 节点进程破坏其他的 ROS 节点进程运行，沙盒技术可以限制可能有危险性的 ROS 节点进程访问磁盘、内存及网络资源。

至于第二个问题，主要原因是通信的信息并没有被加密，以至于攻击者可以轻易地得知通信内容。目前业界有不少对 ROS 节点间通信的加密尝试，比如使用 DES 的加密算法对通信的信息进行加密。在通信的信息量十分小的时候，加密与否对性能影响不大。但是，随着信息量变大，加密时间相对信息量成几何级增长。另外，由于 ROS 通信系统的设计缺陷，加密时间也与接收信息的节点数量有直接关系。当接收信息的节点数量增长时，加密时间也随之增长。我们在第 8 章中提出了几个改进 ROS 通信系统的机制，在这些机制中，加密对性能影响将大大减少。

10.4 无人驾驶控制系统的安全

如图 10-2 所示，车辆的 CAN 总线连接着车内的所有机械及电子控制部件，是车辆的中枢神经。CAN 总线具有布线简单、典型的总线型结构、可最大限度地节约布线与维护成本、稳定可靠、实时、抗干扰能力强、传输距离远等特点。由于 CAN 总线本身只定义 ISO/OSI 模型中的第一层（物理层）和第二层（数据链路层），通常情况下 CAN 总线网络都是独立的网络，所以没有网络层。在实际使用中，用户还需要自己定义应用层的协议，因此在 CAN 总线的发展过程中出现了各种版本的 CAN 应用层协议。CAN 总线采用差分信号传输，通常情况下只需要两根信号线（CAN-H 和 CAN-L）就可以进行正常的通信。在干扰比较强的场合，还需要用到屏蔽地即 CAN-G（主要功能是屏蔽干扰信号）。CAN 总线上任意节点均可在任意时刻主动向其他节点发起通信，节点没有主从之分，但在同一时刻优先级高的节点能获得总线的使用权。

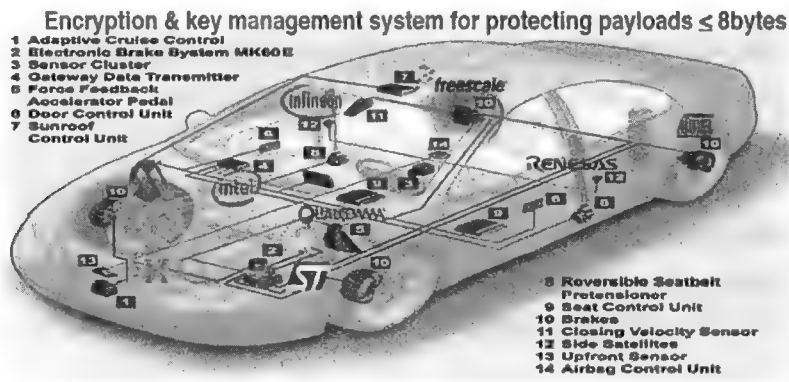


图 10-2 CAN 总线安全

如果 CAN 被劫持，那么黑客将可以为所欲为，造成极其严重的后果。一般来说，要

进入 CAN 系统是极其困难的，但是一般车辆的娱乐系统及检修系统的 OBD-II 端口都连接到 CAN 总线，这就给了黑客进入 CAN 的机会。攻击的方式包括以下几种。

(1) OBD-II 入侵：OBD-II 端口主要用于检测车辆状态，通常在车辆进行检修时，技术人员会使用每个车厂开发的检测软件接入 OBD-II 端口并对汽车进行检测。由于 OBD-II 连接到 CAN 总线，只要黑客取得这些检测软件，包括 Ford's NGS、Nissan's Consult II、Toyota's Diagnostic Tester 等，便可以轻易地截取车辆信息。

(2) 电动车充电器入侵：电动车越来越普及，充电设备也成为了电动车生态必不可少的核心部件。由于电动车的充电装置在充电时会与外部充电桩通信，而且电动车的充电装置会连接 CAN 总线，这就给了黑客通过外部充电桩入侵 CAN 系统的机会。

(3) 车载 CD 机入侵：曾经有攻击的案例是把攻击代码编码到音乐 CD 中，当用户播放 CD 时，恶意攻击代码便会通过 CD 播放器侵入 CAN 总线，从而取得总线控制及盗取车辆核心信息。

(4) 蓝牙入侵：如今蓝牙互连已经成为汽车通信及车上娱乐系统的标配方案。由于我们可以通过蓝牙给 CAN 发送信息及从 CAN 读取信息，也给了黑客攻击的窗口。除了取得车主手机的控制权，由于蓝牙的有效范围是 10 米，黑客也可以使用蓝牙进行远程攻击。

(5) TPMS 入侵：TPMS 是车轮压力管理系统，也有黑客对 TPMS 展开攻击。在这种攻击方法中，黑客先把攻击代码放置在车辆的 TPMS ECU 中，然后当 TPMS 检测到某个胎压值时，恶意代码便会被激活，从而对车辆进行攻击。

如图 10-3 所示，一个通用的解决方法是对 ECU 接收的信息进行加密验证，以保证信息是由可信的 MCU，而不是由黑客发出的。使用加密验证，我们可以选择对称密码，或者非对称密码。对称密码的计算量小但是需要通信双方预先知道密码；非对称密码无须预先知道密码，但是计算量大。由于大部分车用 ECU 计算能力与内存有限，通用的做法是使用对称密码加密，然后密钥在生产过程中被写入 ECU 中。这样的后果是有许多 ECU 复用同一个密钥，当一个 ECU 密钥被破解后，同批的 ECU 都会有风险。为了解决这个问题，学术界和业界也提出了几种解决方案^[8]。

- TLS 安全协议沿用非对称密码的算法对通信双方进行验证。
- Kerberos 是一个通用的基于对称密码的算法验证平台。

- TESLA 安全协议（注意：这个 TESLA 安全协议与 TESLA 汽车没有关系）提出了使用对称密码机制模拟非对称密码的做法，从而达到既安全又能降低计算量的目的。
- LASAN 安全协议使用两步验证的机制让通信双方实时交换密钥，然后使用对称密码的算法对信息进行验证。

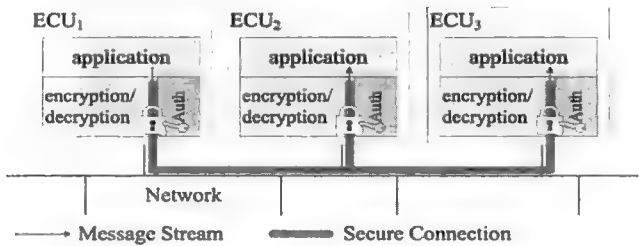


图 10-3 ECU 安全加密系统组成

10.5 车联网通信系统的安全性

当无人车上路后，它会成为车联网的一部分。V2X 是车联网通信机制的总称。可以说，V2X 是泛指各种车辆通信的情景，包括 V2V、V2I、V2P 等。通过 V2X，车辆可以获得实时路况、道路信息、行人信息等一系列交通信息，从而带来远距离环境信号。比如 V2V，最普遍的应用场景是在城市街道、高速公路中，车辆之间可以相互通信，发送数据，实现信息的共享。这些共享数据包括车辆的时速、相对位置、刹车、直行还是左拐等所有这些与行驶安全相关的数据都将提前提供给周围的车辆，使得周围的车辆都能够预判到其他车辆的驾驶行为，从而实现主动的安全策略。V2X 安全防护是无人驾驶必要技术和智慧交通的重要一环，本节我们讨论 V2X 的潜在安全风险及解决方案，如图 10-5 所示。

LTE-V 与 DSRC 是当前车联网的两大技术阵营，前者主要由中国企业推动，后者的发展比 LTE-V 成熟，是欧美等国车联网的主流技术。DSRC 即专用短程通信技术，是基于 IEEE802.11p 标准开发的一种高效的无线通信技术，可提供高速的数据传输，并保证通信链路的低延时和低干扰，可实现小范围内图像、语音和数据的实时、准确和可靠的双向传输。NHTSA（National Highway Traffic Safety Administration）美国国家公路交通安全局已经将 IEEE802.11p 作为 DSRC 的标准协议。



图 10-5 车联网 V2X 系统示意图

802.11p 是在 802.11 的基础上改进的，适应于要求更严格、环境更恶劣的车间通信，其采用 5.9GHz 的频段，通信距离达 300 米。在物理层面，75MHz 被划分为 7 个 10MHz 的信道，频率最低的 5MHz 作为安全空白，中间的一个信道是控制信道，并且有关安全的信息都是广播的形式。边上相邻的两个信道可以用于服务，经过协商后可当作一个 20MHz 的信道使用，比如传输视频之类，其通信优先级别比较低。控制信道使用小点的带宽利于减少多普勒频移效应，两倍警戒间隔减少了多径传输引起的码间干扰。以上改动的结果使物理层的传输速率减少了一半，标准的 802.11p 的传输速率是 3Mbit/s，最大传输速率是 27Mbit/s。室内传输范围 300m，室外最大传输距离 1000m（无阻隔状态）。在芯片级别加强信道管理（对芯片制造商提出更严格的要求），改进 Mac 层让通信工作组更有效率都是 802.11p 的特有之处。SAE（Society of Automotive Engineer）美国汽车工程师协会制定了一组基于 DSRC 的数据消息标准，包括数据内容和帧格式。这个标准的名称为 SAE J2735，其中，这个标准目前的工作重点领域是车间通信的最小性能需求和 BSM（Basic Safety Message，基本安全消息）消息。其中，一部分 BSM 消息需要较高的实时性，需要 1 秒钟发布 10 次，还有另外一些信息，根据实际场景进行广播。DSRC 将需要传输的数据标准化，并将其具体的实现方式也标准化，采用 ASN.1 = Abstract Syntax Notation One，这样从数据消息转换为各种代码实现就有了固定的标准，并有相应的工具支持，免去了各种数据结构的代码定义、编解码实现等。

确保 V2X 通信安全的系统要满足以下两个基本条件：第一，确认消息来自合法的发送设备，这需要通过验证安全证书来保证。第二，确认消息传输过程中没有被修改，这需

要接受信息后计算信息的完整性。为了实现 V2X 的安全, 欧盟发起了 V2X 安全研究项目 PRESERVE, 并在项目中提出了符合 V2X 安全标准的硬件、软件, 以及安全证书架构, 如图 10-6 所示。^[9]

(1) 硬件: 在每个车辆中存储了大量机密的密钥, 如果我们使用普通的 Flash 与 RAM, 密钥会被轻易窃取。另外, 使用加密解密技术会对计算资源消耗极大。为了解决这些问题, PRESEVER 提出了设计安全存储硬件, 以及使用 ASIC 硬件加速加解密。

(2) 软件: 在安全硬件上, PRESEVER 提供了一整套开源软件栈去提供安全通信。这套软件栈提供了加密解密的软件库、电子证书认证库、与受信任的证书颁发机构的安全通信库等。

(3) 安全证书: 为了确保信息来源与可信设备, 我们可以使用受信任的证书颁发机构来提供安全证书与密钥。当汽车 A 向汽车 B 放送信息时, 汽车 A 的发送器会在信息上添加电子签名, 并用密钥对信息进行加密。汽车 B 接收信息时, 会首先对信息的电子证书进行认证, 确认信息是由汽车 A 发送的, 然后使用公钥对信息进行解密, 并对信息的完整性进行验证。

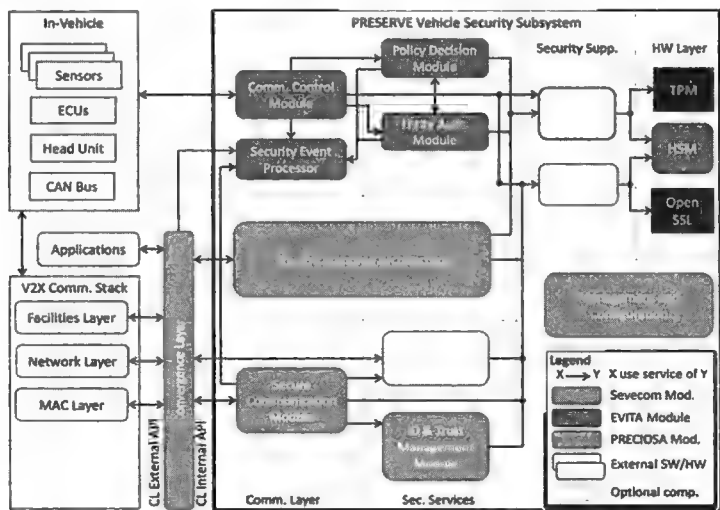


图 10-6 PRESERVE 系统架构图

10.6 安全模型校验方法

为了保证无人驾驶系统的安全性,我们需要从纵向对系统的每个层面进行校验。^{[10][11]}
^{[12][13]} 这些层面包括了代码、电子控制单元 (Electronic Control Unit)、控制算法、车内网及车外网、自动车整体与物理环境结合的所谓网宇实体系统,甚至需要多部车辆互相通信的车联网。越往上层系统的复杂度越大,校验也越困难。所以一般对上层系统的分析会基于下层的分析结果做抽象化处理。例如在分析车内网时,对与网络链接的电子控制单元一般只考虑通信接口的模型,已不会考虑电子控制单元内的具体功能及软件。在对每个层面做安全分析时,我们也需要考虑各种不同的威胁模型和攻击向量。例如,代码的安全校验除了需要考虑缓冲区溢出,还要考虑其他模块通过利用 API 来侵入,或者是第三方软件里载有木马的威胁。在对车内网分析时,要考虑在某个电子控制单元被黑客控制下可能出现的各种情况,包括阻断服务攻击 (Denial of Service Attack)、修改通信件的内容、伪造通信件的来源等。由于无人驾驶系统对处理速度和容量的要求远远高于传统车辆控制系统的需求,一部分单核的电子控制单元在不久的将来会被多核芯片或 GPU 取代。每个新的电子控制单元将会支持多个功能或多个功能的部分实现,而这些功能会通过虚拟机管理硬件资源分配。从安全的角度来说,我们需要对虚拟机管理器进行分析,比如虚拟机与虚拟机之间的通信 (intra-VM communication) 要保证不被第三方干扰或窃听。无人车加入了很多新的自动行驶功能,比如最简单的自动刹车。对于这些功能的控制算法,验证时我们也需要全面地考虑前文所提到的一系列威胁,包括某个传感器的信息被恶意修改、通信渠道被堵所引起的信息滞后等。因为无人车需要强大的 AI 系统做支持,对这些 AI 系统的不同攻击方式也在校验的考虑范围内。最近有研究指出,深度学习系统 (应用在图像识别上) 也很容易被攻击。例如,修改一张图像中的几个像素就可能使识别结果大相径庭。这个隐患大大增加了系统被黑客攻破的可能性。在车联网的层面上,常见的安全问题有通信信息被篡改,被黑客控制的车辆故意提供假信息或伪造身份,阻断服务攻击,女巫攻击 (sybil attack: 单辆车通过控制多个身份标识对网络整体进行攻击), 以及盗取其他车主的私密信息 (比如所在位置)。

对于这些安全问题及攻击向量的分析涉及的技术非常广。本章,我们重点介绍了关于车内网 (比如前面提到的 CAN) 和控制系统的的核心模型和验证。现有的车内网安全协议一般建立在一些基本的加密单元上,比如对称密钥加密和非对称密钥加密。一般初始身份鉴别时需要用非对称密钥加密,而之后的通信就可以用相对更快的对称密钥加密。根据不同的安全等级需求,密钥的长度会不一样。长的密钥会更安全,也会增加加密和解密的时间。

间,因此影响到控制系统的性能。另外,长的密钥会增加通信的负担。不管是 CAN 还是 TDMA 类的车内网协议,这些附加的安全信息都可能导致通信超时(结果可能是来不及刹车)。所以在安全校验的同时也必须考虑增加安全机制所产生的延时。最后,密钥的分发和管理也至关重要。这是当前的一个技术难点,还没有特别好的解决方案。对于协议本身的验证方法有几种。一般来说,我们先校验协议的数学模型。最近提出的 LASAN 就是先用形式化验证工具 **Scyther** 证明协议的安全性,然后做仿真来测试性能。对于控制系统,分析时是会侧重考虑攻击对数据所产生的影响(比如延时、丢失或假数据),然后对相应的安全方案(比如传感器数据混合处理或状态估计)做数学证明来达到校验的目的。类似的方法也被应用在验证一些车联网的功能上,像合作的可变巡航控制。总体来说,无人车的安全问题至关重要,如果车辆被黑客攻击或控制会危及生命。但是,不管从技术还是标准化的角度看,现阶段对于无人车安全问题的校验尚未成熟,还需要学术界和工业界的深入研究与大力开发。

10.7 参考资料

- [1] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno et al., "Comprehensive experimental analyses of automotive attack surfaces", in *Proc. 20th USENIX Security*, San Francisco, CA, 2011.
- [2] Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H. and Savage, S., 2010, May. Experimental security analysis of a modern automobile. In *Security and Privacy (SP)*, 2010 IEEE Symposium on (pp. 447-462). IEEE.
- [3] Dominic, D., Chhawri, S., Eustice, R.M., Ma, D. and Weimerskirch, A., 2016, October. Risk Assessment for Cooperative Automated Driving. In *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy* (pp. 47-58).ACM.
- [4] Petit, J., Stottelaar, B., Feiri, M. and Kargl, F., 2015. Remote attacks on automated vehicles sensors: Experiments on camera and lidar. *Black Hat Europe*, 11, p.2015.

- [5] Shoukry, Y., Nuzzo, P., Puggelli, A., Sangiovanni-Vincentelli, A.L., Seshia, S.A. and Tabuada, P., 2014. Secure state estimation for cyber physical systems under sensor attacks: a satisfiability modulo theory approach. arXiv preprint arXiv:1412.4324.
- [6] Nguyen, A., Yosinski, J. and Clune, J., 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 427-436).
- [7] Lera, F.J.R., Balsa, J., Casado, F., Fernández, C., Rico, F.M. and Matellán, V., 2016. Cybersecurity in Autonomous Systems: Evaluating the performance of hardening ROS. Málaga, Spain, p.47.
- [8] Mundhenk, P., Steinhorst, S., Lukasiewicz, M., Fahmy, S.A. and Chakraborty, S., 2015, March. Lightweight authentication for secure automotive networks. *In Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition* (pp. 285-288). EDA Consortium.
- [9] Bißmeyer, N., Petit, J., Estor, D., Sall, M., Stotz, J.P., Feiri, M., Moalla, R. and Dietzel, S., 2011. PRESERVE d1. 2 v2x security architecture.Deliverable, PRESERVE consortium.
- [10] Seshia, S.A., Sadigh, D. and Sastry, S.S., 2015, June. Formal methods for semi-autonomous driving. *In Proceedings of the 52nd Annual Design Automation Conference* (p. 148).ACM.
- [11] Wongpiromsarn, T. and Murray, R.M., 2008, May. Formal verification of an autonomous vehicle system. *In Conference on Decision and Control*.
- [12] Clarke, E., Garlan, D., Krogh, B., Simmons, R. and Wing, J., 2001. Formal verification of autonomous systems NASA intelligent systems program.
- [13] Simmons, R., Pecheur, C. and Srinivasan, G., 2000. Towards automatic verification of autonomous systems. *In Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings.2000 IEEE/RSJ International Conference on* (Vol. 2, pp. 1410-1415). IEEE.

基于 Spark 与 ROS 的分布式无人驾驶模拟平台

本章着重介绍基于 Spark 与 ROS 的分布式无人驾驶模拟平台。无人驾驶的安全性和可靠性是通过海量的功能和性能测试保证的。无人驾驶系统是一个复杂的系统工程, 在整个研发流程中, 测试工作至关重要同时也繁重复杂。显然全部测试工作都集中在真车上进行是一种成本异常高昂, 且安全系数非常低的测试方案。通过综合考虑测试中各种可能发生的正常或异常状况, 软件模拟成为了面向无人驾驶系统的更安全且更经济有效的替代测试手段。

11.1 无人驾驶模拟技术

无人车驾驶系统由感知、预测、决策、控制等众多功能模块组成, 每个模块都各自拥有复杂的结构和算法。绝大部分情况下, 在测试过程中系统开发人员很难对海量的输出参数做评价。同时, 开发人员不仅需要单独测试一个功能模块, 也需要集合联调多个模块。因此, 系统开发人员需要的模拟器必须能够直观正确地反映出输出参数的意义, 同时既能对各个模块进行单一的集成测试, 又能将各个模块按照需求分别组合后进行集成测试。^[1]

模拟器技术主要有两种: 第一种是基于合成数据, 对环境、感知及车辆进行模拟, 这

种模拟器主要用于控制与规划算法的初步开发上；第二种是基于真实数据的回放以测试无人驾驶不同部件的功能及性能。本节，我们主要讨论基于数据回放的模拟器。

出于尽量真实地模拟真车环境的需求，我们的模拟器采用了和真车相同的机器人操作系统 ROS。ROS 是一种基于消息传递通信的分布式计算框架。这种框架方便开发人员行模块化编程，这一特性对于模拟器来说至关重要。在无人驾驶系统中，每一个功能模块在 ROS 中都部署在一个节点上，节点间的通信依靠事先定义好格式的 `message` 完成。在模拟器中开发人员只需要使用相同的通信格式，针对每个功能模块制作模拟模块，就可以根据测试需求搭配真实功能模块和模拟模块。例如，如果想进行决策模块和控制模块的功能联调，我们需要将决策模块、控制模块搭配其他的模拟模块，并安装到模拟器中进行测试。如果决策模块需要单独测试新的决策算法，我们可以只将新的决策模块搭配其他的模拟模块安装到模拟器上，这样的测试结果只是针对决策模块的。

11.1.1 模拟器的组成元素

首先，无人车模拟器中包含的是车的动态模型。车的动态模型是用来加载测试无人车驾驶系统，并模拟无人车自身的行为的。其次，模拟器需要模拟的是外部环境。外部环境主要分为静态的场景和动态的场景。静态场景中包括各种静态的交通标志，例如停止线、交通指示牌等。动态场景主要指车周围的动态交通流模型，例如车辆、行人、交通灯等。所有这些元素构建了与现实环境相对应的模拟世界。

11.1.2 模拟器的应用

无人车真实上路后要面临的外部环境是复杂多变的。模拟器在模拟测试中需要做的就是将复杂的外部环境拆解成最简单的元素，然后重新排列组合，生成各种测试用例。

拿一组简单的测试用例举例。图 11-1 所示为一个简单的直线行驶的车道，需要测试的是无人车对于一辆障碍车的反应。按照障碍车可能出现的起始位置划分，障碍车可能出现在无人车的左前、左中、左后、前、后、右前、右中、右后总计八个位置。按照障碍车和无人车的相对速度划分，障碍车分为比无人车快、与无人车速度相等、比无人车慢这三类。按照障碍车的行为划分则分为直行、向左变道和向右变道三种。将这些变量相乘，去掉其中不需要的个例就得到了一组我们需要的测试用例。



图 11-1 模拟器在无人驾驶中的应用

11.1.3 模拟器面临的问题

模拟器的核心问题在于“真”上，人工模拟的场景和真实场景多少会有差异，真实场景中仍然会存在许多人想象不到的突发事件。因此，如果能采用真实的行车数据复现真实场景将会得到比人工模拟的场景更好的测试效果。但采用真实数据复现的方案随之带来的问题就是海量数据的处理。如果我们想在模拟器上复现真实世界中每一段道路的场景，我们就需要让无人车采集每一段道路的信息，这些海量的信息是单机无法处理的，而且在每个场景下拆解元素重新排列组合生成测试用例的做法会使计算量翻倍。因此，将模拟器搭载到分布式系统上就成为了无人驾驶模拟测试的最佳选择。

11.2 基于 ROS 的无人驾驶模拟器

ROS 是一种基于消息传递通信的分布式计算框架。它的通信模式可以抽象为一种 message pool 的架构，消息发送节点调用 `advertise` 方法向指定 Topic 发送 ROS message，消息接收节点调用 `subscribe` 方法从指定 Topic 接收 ROS message。^[2]

11.2.1 Rosbag

Rosbag 是一套利用这一架构从 Topic 中录制并且向 Topic 重新播放 ROS message 的工具，在无人车的数据采集过程中，使用的正是 Rosbag 这套工具。Rosbag 的功能主要分为 Record 和 Play 两类。Record 功能是在 ROS 中建立一个 record 节点，调用 `subscribe` 方法向所有或指定 Topic 接收 ROS message，然后将 message 写入 Bag 文件。Play 功能则是在 ROS 中建立一个 play 节点，调用 `advertise` 方法将 bag 中的 message 按照时间节点发送至指定 Topic。图 11-2 所示为一个 LiDAR 数据在 ROS 中回放的实例，在这个场景中，LiDAR 数据是以 10Hz 的帧率记录的。

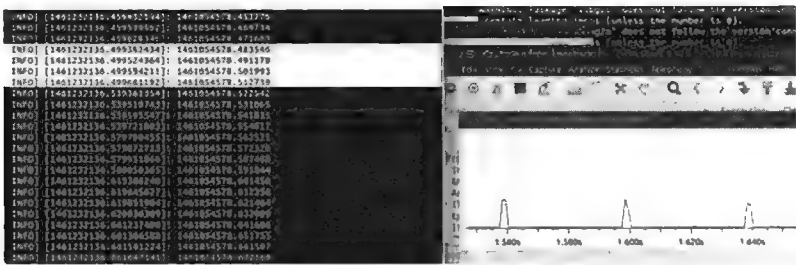


图 11-2 Ros bag LiDAR 数据回放原理

Rosbag 生成的数据格式是 Bag，这是一个拥有两层逻辑结构的文件格式。如图 11-3 所示，上层的 Bag 类对上抽象提供了用户操作文件的方法，对下封装了对 ChunkedFile 的操作方法，ChunkedFile 类主要对数据进行了分隔存储，而存储的数据为一条条的 ROS message，ROS message 的内容不仅包含文字信息，有时也包含大量的二进制数据，这些主要是无人车的 sensor 发送的图片或者 3D 点云文件的数据。这就给传统的主要用来处理文字日志的分布式计算系统应用带来了新的挑战。



图 11-3 Rosbag 结构图

11.2.2 模拟测试数据集

如上所述，我们主要关注基于真实数据回放的模拟器，那么这个数据量有多大呢？我们将以 KITTI 数据集为实例了解一下。KITTI 数据集是由 KIT 和 TTIC 在 2012 年开始的一个合作项目，网址为 <http://www.cvlibs.net/datasets/kitti/>，这个项目的主要目的是建立一个具有挑战性的、来自真实世界的测试集。他们使用的数据采集车配备了一对 140 万像素的彩色摄像头、Point Grey Flea 2 (FL2-14S3C-C)、10Hz 采集频率；一对 140 万像素的黑白摄像头、Point Grey Flea 2 (FL2-14S3M-C)、10Hz 采集频率；一个激光雷达、Velodyne HDL-64E；一个 GPS/IMU 定位系统、OXTS RT 3003。^[4]

KITTI 的研究人员使用这个配置录制了 6 个小时的真实数据,数据量为 720GB,但是 6 小时的数据仅够完成一些算法的简单验证,而无人驾驶产品所需求的数据远大于此。比如谷歌的无人车在过去几年中收集了超过 40000 小时的真实数据,总数据量应该超过了 5PB。这样基于单机的模拟远不能支撑如此大量的数据处理,所以我们必须为基于真实数据回放的模拟器设计一个高效的分布式计算平台。

11.2.3 计算量的挑战

巨大的数据处理量对计算平台造成了很大的压力。例如, KITTI 数据集整 6 小时的原数据包括了超过 1000000 张 140 万像素的彩图,使用我们的基于深度学习的图像识别平台(单机),每张彩图分析时间大概是 0.3 秒。这样,仅是分析 KITTI 数据集的图片,在单机上就需要超过 100 小时,而如果分析谷歌无人车级别的整体图片数据,在单机上需要超过 60 万个小时。

11.3 基于 Spark 的分布式的模拟平台

Spark 是 UC Berkeley AMP Lab 开源的通用并行计算框架。Spark 基于内存实现的分布式计算,拥有 Hadoop 所具有的优点;但不同于 Hadoop, Spark Job 的中间输出和结果可以保存在内存中,从而不再需要读写 HDFS,因此 Spark 能更好地适用于需要迭代的 map-reduce 算法。^[3]

如图 14-4 所示,为了高效的分布式的进行无人驾驶回放模拟,我们设计了基于 Spark 的分布式模拟平台框架。我们使用 Spark 进行资源的分配管理、数据的读写,以及 ROS 的节点管理。在 Spark Driver 上,我们可以触发不同的模拟应用,比如基于 LiDAR 的定位、基于图片的物体识别、车辆决策与控制等。Spark Driver 会根据数据量与计算量等需求请求 Spark Worker 资源。每个 Spark Worker 首先会把 Rosbag 数据读入内存,然后通过 pipe 启动 ROS Node 进程进行计算。我们也可以使用 JNI 方式连接 Spark Worker 及 ROS Node,但是这样将涉及对 ROS 的修改,使得整个系统难以维护与迭代。在权衡过后,我们最终选择了 pipe 的设计方案,如图 11-4 所示。

在 pipe 的设计方案中,有两个问题需要解决:第一, Spark 本身支持文本数据读取,但并不支持多媒体数据读取,我们需要设计一个高效的二进制文件的读取方法。第二, Rosbag 的 play 功能如何从内存中读取缓存的数据,record 功能如何将数据缓存至内存中。

我们将讨论以下这些设计。

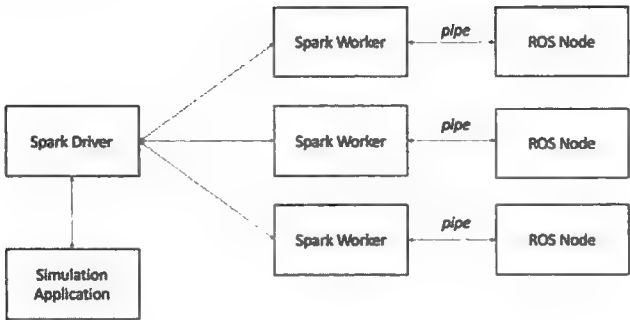


图 11-4 分布式模拟平台总体架构图

11.3.1 二进制文件流式管道处理

Spark 操作数据的核心是弹性分布式数据集（RDD），它允许程序员以一种容错的方式在一个大型集群上执行内存计算。百度公司美国研发中心之前的一个工作就是在这—数据结构的基础上引入一个新的 RDD 来实现二进制文件流式管道处理。其结构如图 11-5 所示。

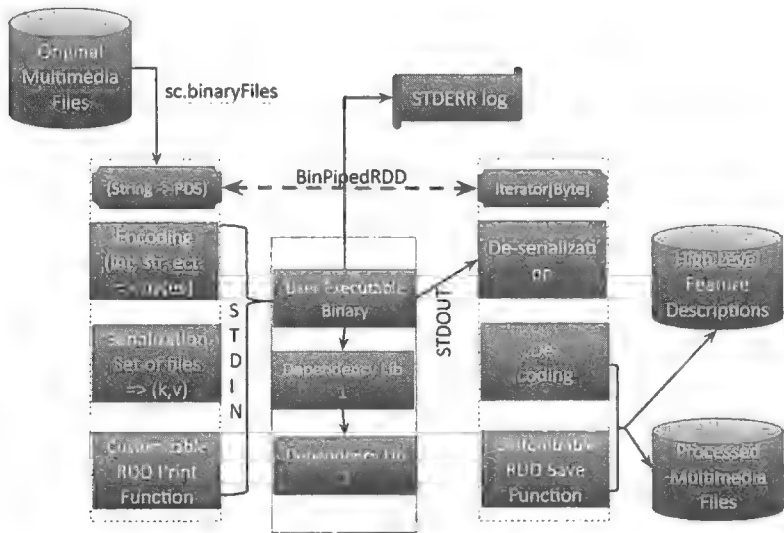


图 11-5 BinPipedRDD 的总体设计和主要功能示意图

在每一个 Spark 的 worker 上，worker 根据 BinpipedRDD 的信息通过标准输入流在内

存中将数据传送给用户程序,用户程序处理完数据后通过标准输出流在内存中将数据传回给 Spark 的 worker。worker 将数据汇集存储到 HDFS 上。

11.3.2 Rosbag 缓存数据读取

在当前使用场景下(如图 11-6 所示),我们的输入是一定量的 Bag 二进制文件(文件以某种形式存储在分布式文件系统上面),而用户想要的输出是所有这些 Bag 文件在每一个 worker 上回放信息进入模拟器后经过处理得到的数据,显然这一过程通过 Rosbag 的 play 和 record 功能最易实现。

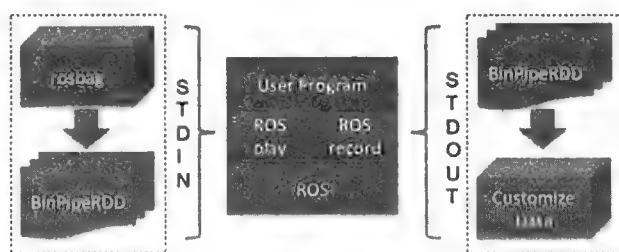


图 11-6 模拟器在分布式平台的运作流程图

不过这一过程中还存在缺失的环节,即 Rosbag 的 play 功能如何从内存中读取缓存的数据,record 功能如何将数据缓存至内存中。为了实现这一功能,我们为原来的 Bag 和 ChunkedFile 的两层逻辑结构增加了一个分支逻辑层。如图 11-7 所示,MemoryChunkedFile 类继承于 ChunkedFile 类并且重写了 ChunkedFile 所有的方法。MemoryChunkedFile 在向底层读写文件时是向内存读写数据,并不是像 ChunkedFile 类一样向硬盘读写数据。这样做带来了一个好处就是 worker 通过标准输入流传给模拟器的数据不用经过磁盘 IO 读写就可以被直接读入,经过模拟器处理的数据不用经过磁盘 IO 读写就可以由内存直接传回 worker。这样的读写模式极大地减少了模拟器处理数据的时间。

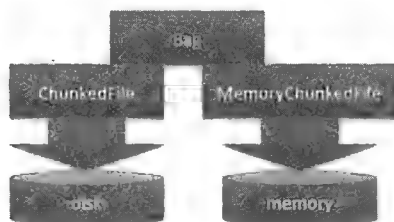


图 11-7 MemoryChunkedFile 结构设计

通过这一逻辑层的添加，我们可以将模拟器部署到 Spark 机群内的每一台 worker 机器上。我们可以通过加载不同的配置文件使每台机器运行不同的模块；也可以通过部署相同模块不同模型、运行相同数据以比较模型的不同；还可以在相同模块相同模型的条件运行不同数据，以比对不同数据的差异。由此可见，分布式系统的使用为模拟器添加了无数扩展的可能性。

11.3.3 性能评估

在设计实现的同时，我们对系统进行了性能评估。随着计算资源的增加，计算时间也在线性降低，系统表现出很强的可扩展性，可以承受很大的数据量与计算量。在一个图像识别测试集中，使用单机处理的图像数据耗时为 3 个小时，而使用 8 个 Spark Worker 后，耗时仅仅为 25 分钟。假设我们使用 10000 个 Spark Worker 对谷歌无人车级别的数据进行大规模的图像识别模拟测试，那么整个实验可以在 100 小时内完成。

11.4 结论

使用分布式系统能够极大提升模拟器的工作能力，使得无人驾驶系统的测试工作得以大规模有序化的扩展。这一结果是建立在模拟器架构模块化，以及测试用例组合模块化的基础之上的。采用分布式系统搭建模拟平台使得在真车上路之前，测试无人车将行驶的每一条道路成为现实。当然，无人车在真实道路上的测试依然必不可少，但是模拟器已经为无人驾驶系统测试了海量的基础情景，可以以最低的成本来最大限度地保障真车测试时的安全性。

11.5 参考资料

- [1] Krogh, B. and Thorpe, C., 1986, April. Integrated path planning and dynamic steering control for autonomous vehicles. *In Robotics and Automation. Proceedings. 1986 IEEE International Conference on* (Vol. 3, pp. 1664-1669).IEEE.
- [2] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. and Ng, A.Y., 2009, May. *ROS: an open-source Robot Operating System*. In *ICRA workshop on open source software* (Vol. 3, No. 3.2, p. 5).

- [3] Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S. and Stoica, I., 2010. Spark: Cluster Computing with Working Sets. *HotCloud*, 10(10-10), p.95.”
- [4] Geiger, A., Lenz, P., Stiller, C. and Urtasun, R., 2013. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11), pp.1231-1237.

无人驾驶中的高精度地图

高精度地图作为无人驾驶发展成熟标志的重要支撑，在横向/纵向精确定位、基于车道模型的碰撞避让、障碍物检测和避让、智能调速、转向和引导等方面发挥着重要作用，是无人驾驶的核心技术之一。精准的地图对无人车的定位、导航与控制，以及无人驾驶的安全至关重要。本章首先介绍高精地图与传统地图的区别，然后介绍高精地图的特点及制作过程。在了解了高精地图的基础知识后，我们探索高精地图在无人驾驶场景中的应用。

12.1 电子地图分类

我们日常使用的用于导航、查询地理信息的地图都属于传统电子地图，其主要服务对象是人类驾驶员。与传统电子地图不同，高精度电子地图的主要服务对象是无人驾驶系统。本节讨论传统电子地图与高精度电子地图的不同。

12.1.1 传统电子地图

如图 12-1 所示，我们日常使用的用于导航、查询地理信息的地图，如谷歌地图、百度地图、苹果地图等，都可以并入传统电子地图。尽管电子地图出现还不到一百年，对传统地图的研究和开发已经有几千年的历史，并发展出制图学这一门学科。在制图学的基础上，电子地图的出现极大地提高了地图的检索效率，并且能快速地查找最优路径，极大地方便了人们的出行。



图 12-1 传统电子地图示意图

传统电子地图是对路网的一种抽象：所有的传统地图都将路网抽象成有向图的形式：图的顶点代表路口，边代表路口与路口的连接。路名、地标及道路骨架信息都可以被抽象成存储于这些有向图顶点或边中的属性。这种抽象的地图表征形式能很好地适应人类驾驶员的需求，其原因在于人类生来就有很强的视觉识别及逻辑分析能力。在驾驶的过程中，人类驾驶员一般都能有效地判别如下信息：识别路面及路面标示线，确定自己在路面的大致位置，寻找并辨认路标，等等。参照这些辨识出的信息，结合当前 GPS（一般的精度在 5~10 米）在当前电子地图中的位置，人类驾驶员便可大致知道自己在实际路网中的位置，并计划下一步如何驾驶。正是基于人类驾驶员的这些能力，传统的电子地图可以被极大精简，比如一条弯曲的道路可以被精简到用只有几个点的线段来表示，只要大致的轮廓符合现实路网的结构，人类驾驶员即可结合当前的驾驶信息定位自己的当前位置。

12.1.2 高精度电子地图

与传统电子地图不同，高精度电子地图的主要服务对象是无人车，或者说是机器驾驶员。和人类驾驶员不同，机器驾驶员缺乏人类与生俱来的视觉识别、逻辑分析的能力。例如，人可以很轻松、准确地利用图像、GPS 定位自己，鉴别障碍物、人、交通信号灯等，但这些对当前的机器人来说都是非常困难的任务。借助高精度地图能够扩展车辆的静态环境感知能力，为车辆提供其他传感器提供不了的全局视野，包括传感器监测范围外的道路、交通和设施信息。高精度地图面向无人驾驶环境采集生成地图数据，根据无人驾驶需求建立道路环境模型，在精确定位、基于车道模型的碰撞避让、障碍物检测和避让、智能调速、

转向和引导等方面都可以发挥重要作用，是当前无人车技术中必不可少的一个组成部分，如图 12-2 所示。

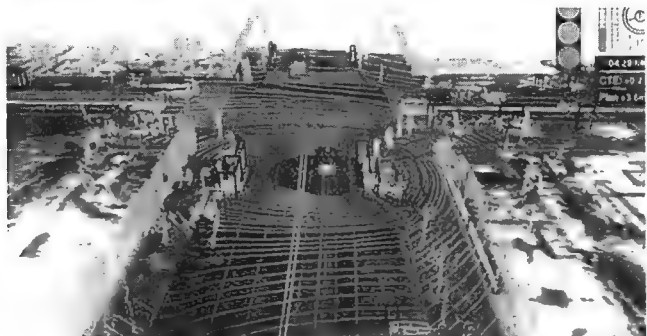


图 12-2 高精度电子地图示意图

高精度电子地图包含大量的行车辅助信息。如图 12-3 所示，这些辅助信息可以分成两类，一类是道路数据，如道路车道线的位置、类型、宽度、坡度和曲率等车道信息；另一类是行车道路周围相关的固定对象信息，如交通标志、交通信号灯等信息、车道限高、下水道口、障碍物及其他道路细节，还包括高架物体、防护栏、树、道路边缘类型、路边地标等基础设施信息。所有上述信息都有地理编码，因此导航系统可以准确定位地形、物体和道路轮廓，从而引导车辆行驶。其中最重要的是对路网精确的三维表征(厘米级精度)，例如路面的几何结构、道路标示线的位置、周边道路环境的点云模型等。有了这些高精度的三维表征，车载机器人就可以通过比对车载的 GPS、IMU、LiDAR 或摄像头的数据精确地确认自己当前的位置。除此以外，高精度地图还包含丰富的语义信息，比如交通信号灯的位置及类型、道路标示线的类型、识别哪些路面是可以行驶的，等等。通过对高精度地图模型的提取，可将车辆位置周边的道路、交通、基础设施等对象及对象之间的相对关系提取出来。这些能极大地提高车载机器人鉴别周围环境的能力。此外，高精度地图还能帮助无人车识别车辆、行人及未知障碍物，因为一般的地图会过滤掉车辆、行人等活动障碍物。如果无人车在行驶过程中发现在当前高精度地图中没有的物体，这些物体有很大的概率是车辆、行人或障碍物。因此，高精度地图可以提高无人车发现并鉴别障碍物的速度和精度。^[1]

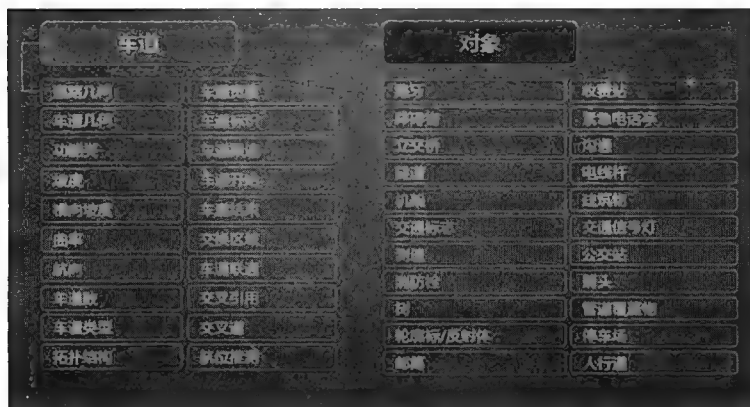


图 12-3 高精度地图识别对象

12.2 高精度地图的特点

相比服务于 GPS 导航系统的传统地图而言，高精度地图最显著的特征是其表征路面特征的精准性。一般情况下，传统地图只需要做到米量级的精度即可实现基于 GPS 的导航，但高精度地图需要至少 10 倍以上的精度，即达到厘米级的精度才能保证无人车行驶的安全。目前，商用 GPS 的精度仅有 5 米左右，而高精度地图与传感器协同工作，可将车辆的位置定位精确到厘米级(TomTom、Google、Here 等产品的定位精度都达到 10~20 厘米)。

此外，高精度地图还需要有比传统地图更高的实时性。由于道路路网每天都会有变化，比如道路整修、道路标识线磨损及重漆、交通标示改变等。这些改变需要及时反映在高精度地图上以确保无人车行驶安全。要做到实时的高精度地图有很高的难度，但随着越来越多载有多种传感器的无人车行驶在路网中，一旦有一辆或几辆无人车发现了路网的变化，通过和云端的通信，就可以把路网更新信息告诉其他的无人车，使得其他无人车变得更加聪明和安全。

12.2.1 数据特征类型

和传统地图相似，高精度地图也具有分层的数据结构。如图 12-4 所示，底层是一个基于红外线雷达传感器建立的精密二维网格。一般这个二维网格的精度保证在 5×5 厘米左右，可以行驶的路面、路面障碍物，以及路面在激光雷达下的反光强度都被存储于相应

的网格中。无人车在行驶的过程中,通过比对其红外线雷达搜集到的数据及其内存中的高精二维网格,就能确定车辆在路面的具体位置。



图 12-4 高精度地图二维网格表征

除了底层的二维网格表征外,高精度地图还包含很多有关路面的语义信息。如图 12-5 所示,在二维网格参照系的基础上,高精度地图一般还包含道路标识线的位置及特征信息,以及相应的车道特征。由于车载的传感器可能会因为恶劣天气、障碍物,以及其他车辆的遮挡不能很可靠地分析出车道信息,高精度地图中的车道信息特征能帮助无人车更准确可靠的识别道路标识线,并理解相邻车道之间是否可以安全并道。

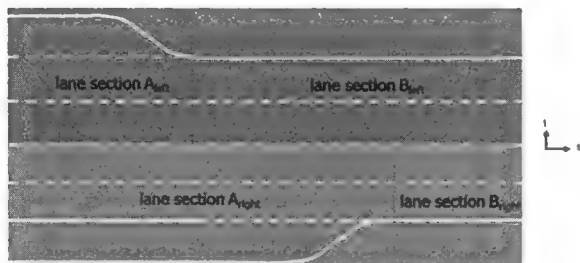


图 12-5 高精度地图中的车道信息

此外,如图 12-6 所示,高精度地图还会标明道路标示牌、交通信号等相对于二维网格的位置。这些信息有如下两方面的作用。

(1) 提前预备无人车,告诉无人车在某些特定的位置检测相应的交通标示牌或交通信号灯,提高无人车的检测速度。

(2) 在无人车没有成功检测出交通标示牌或信号灯的情况下,确保行车的安全。

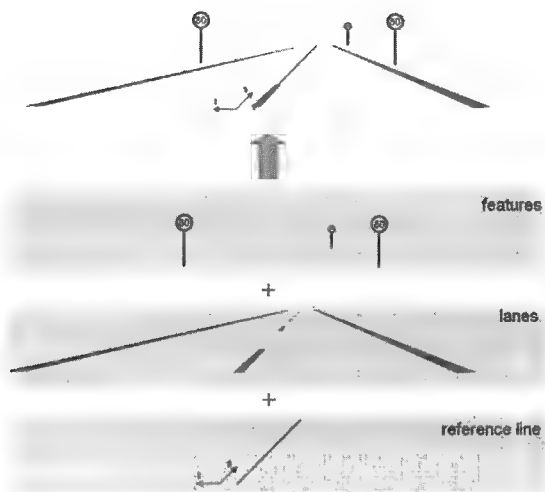


图 12-6 高精度地图中的道路标识线及路牌信息

12.2.2 数据量估计

无人车使用的高精度地图是个 2D 的网格，数据主要由激光雷达产生，由于激光雷达的精度大约是 5 厘米左右，所以地图的最高精度可以达到每个网格 5 厘米×5 厘米。在如此高的精度下，如何有效地管理数据是高精度地图的一个大挑战。首先，为了尽量让地图在内存里面，我们要尽量去掉不需要的数据。一般的激光雷达可覆盖方圆 100 米的范围，假设每个反光强度可以用一个字节记录，那么每一次激光雷达扫描可以产生 4MB 的数据。这样的扫描会包括公路旁边的树木及房屋，但是无人车的行驶并不需要这些数据，我们只需要记录公路表面的数据即可。假设路面的宽度为 20 米，那么我们可以通过数据处理把非公路表面的数据过滤掉，这样每次扫描的数据量会下降到 0.8MB。在过滤数据的基础上，我们可以使用无损的压缩算法，比如 LASzip 去压缩地图数据，可以达到超过 10 倍的压缩率。经过这些处理后，一个 1TB 的硬盘就可以存下全中国超过 10 万公里的高精地图数据。

12.3 高精度地图的生产

传统的电子地图主要依靠卫星图片产生，然后依靠 GPS 定位，这种方法可以达到米级精度，而高精地图需要达到厘米级精度，仅仅靠卫星与 GPS 是不够的。因此，高精地

图的生产涉及多种传感器，由于产生的数据量很大，通常会使用数据采集车（如图 12-7 所示）收集数据，然后通过线下处理把各种数据融合产生高精地图。



图 12-7 配备各种传感器的高精度地图数据采集车

12.3.1 高精度地图需要的传感器种类

高精度地图的制作是一个多传感器融合的过程，包括以下传感器。

（1）陀螺仪（IMU）：一般会使用 6 轴运动处理组件，包含了 3 轴加速度和 3 轴陀螺仪。加速度传感器是力传感器，用来检查上、下、左、右、前、后哪几个面受了多少力（包括重力），然后计算每个轴 $\langle x, y, z \rangle$ 上的加速度。陀螺仪就是角速度检测仪，检测每个轴 $\langle x, y, z \rangle$ 上的角速度。假设一辆无人车以 Z 轴为轴心，在一秒钟的时间转到了 90° ，那么它在 Z 轴上的角速度就是 $90^\circ/\text{秒}$ 。从加速度推算出运动距离需要经过两次积分，所以，但凡加速度测量上有任何不正确，在两次积分后，位置的错误会积累然后导致位置预测错误。所以，单靠陀螺仪并不可以精准地预测无人车的位置。

（2）轮测距器（Wheel Odometer）：我们可以通过轮测距器推算无人车的位置。在汽车的前轮通常安装了轮测距器，会分别记录左轮与右轮的总转数。通过分析每个时间段里左右轮的转数，我们可以推算出车辆向前走了多远，向左右转了多少度等。可是由于在不同地面材质（比如冰面与水泥地）上转数对距离转换的偏差，随着时间的推进，轮测距的测量偏差会越来越大。所以单靠轮测距器并不可以精准地预测无人车位置。

（3）GPS：GPS 接收机的任务就是确定四颗或更多卫星的位置，并计算出它与每颗卫星之间的距离，然后用这些信息使用三维空间的三边测量法推算出自己的位置。要使用距离信息进行定位，接收机还必须知道卫星的确切位置。GPS 接收机存储有星历，其作用是告诉接收机每颗卫星在各个时刻的位置。在无人车复杂的动态环境，尤其在大城市中，

由于各种高大建筑物的阻拦。GPS 多路径反射 (Multi-Path) 的问题会更明显。这样得到的 GPS 定位信息很容易就有几十厘米甚至几米的误差, 所以单靠 GPS 不可以制作高精地图。

(4) LiDAR: 光学雷达通过首先向目标物体发射一束激光, 然后根据接收—反射的时间间隔确定目标物体的实际距离。根据距离及激光发射的角度, 通过简单的几何变化可以推导出物体的位置信息。LiDAR 系统一般分为三个部分: 第一是激光发射器, 发射出波长为 600nm~1000nm 的激光射线; 第二部分是扫描与光学部件, 主要用于收集反射点距离与该点发生的时间和水平角度 (Azimuth); 第三个部分是感光部件, 主要检测返回光的强度。因此, 我们检测到的每一个点都包括了空间坐标信息 $\langle x, y, z \rangle$ 及光强度信息 $\langle i \rangle$ 。光强度与物体的光反射度 (reflectivity) 直接相关, 所以从检测到的光强度我们也可以对检测到的物体有初步判断。

12.3.2 高精度地图计算模型

图 12-8 所示为一个通用的高精地图制作流程, 过程中涉及了多个传感器与多个计算步骤。首先, 陀螺仪 (IMU) 及轮测距器 (Wheel Odometer) 可以高频率地给出当前无人车的位置预测, 但是由于陀螺仪及轮测距器的精确度原因, 给出的位置可能会有一定程度的偏差。为了纠正这些偏差, 我们可以使用传感器融合计技术 (比如使用 Kalman Filter) 结合 GPS 与 LiDAR 的数据算出当前无人车的准确位置。然后根据当前的准确位置与激光雷达的扫描数据, 我们可以把新的数据加入地图中^[3]。

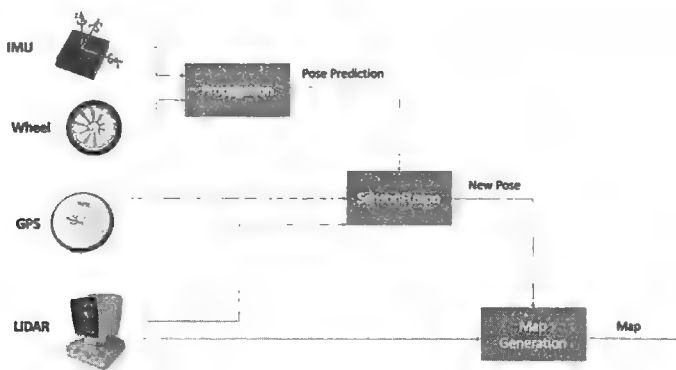


图 12-8 高精地图计算架构示意图

下面的公式是个高度简化的高精地图计算模型, Q 代表优化方程, z 代表激光雷达扫

描出的点, h 方程预测最新扫描点的位置与反光度, m 扫描到的点在地图中的位置, x 代表无人车当前的位置。这个方程的目的是通过最小化 J 求出测量出的点在地图中的准确位置。在这个计算模型中, m 与 x 开始都是未知的, 所以我们可以先通过多传感器的融合求出无人车的位置 x , 然后求出测量点在地图中的准确位置 m 。

$$J = Q(z - h(m, x))$$

12.4 无人驾驶场景中的应用

高精度电子地图包含大量的行车辅助信息, 包括路面的几何结构, 道路标示线的位置, 周边道路环境的点云模型等。有了这些高精度的三维表征, 无人驾驶系统就可以通过比对车载的 GPS、IMU、LiDAR 或摄像头的数据精确地确认自己当前的位置, 并进行实时导航。

12.4.1 定位

无人车对可靠性和安全性的要求非常高, 所以这里默认我们有高精度 LIDAR 和多种传感器融合建好了地图, 在这个前提下, 我们谈一谈无人车的跟踪和定位技术。^{[2][4]}

无人车定位主要是通过粒子滤波进行的。所谓粒子滤波就是指通过寻找一组在状态空间中传播的随机样本来近似地表示概率密度函数, 用样本均值代替积分运算, 进而获得系统状态的最小方差估计的过程, 这些样本被形象的称为“粒子”, 故而叫粒子滤波。比较常见的 (比如在 Sebastian Thrun 的经典的无人车论文里) 是粒子滤波维护一个姿态向量 (x, y, yaw) , 默认 roll/pitch 相对足够准, 运动的预测可以从 IMU 里拿到加速度和角速度。粒子滤波需要注意样本贫化和其他可能的灾难定位错误 (catastrophic error), 一小部分粒子可以持续从现在 GPS 的位置估计中获得。对样本数量的自适应控制也需要根据实际情况有效调整。

因为已经有了高精度的 LIDAR 点云地图, 所以很自然就可以用实时的 LiDAR 点云数据和已经建好的地图进行匹配。而 3D 点云的匹配必然要说到 Iterative Closest Point (ICP), ICP 的目标是给出两组点云的情况下, 假设场景不变, 算出这两组点云之间的 pose。最早的 ICP 的道理简单说就是第一组点云的每一个点在第二组点云里找到一个最近的匹配, 之后通过所有的匹配计算均方误差 (MSE), 进而调整估计的 pose, 这样进行多次迭代最终算成两组点云的相对 pose。所以预先有了地图的情况下用实时的点云加上一个大

概的 **pose** 猜测就可以精准地算出无人车当前的 **pose**，而且时间上相邻的两帧点云也可以算出一个相对的 **pose**。

另一方面，因为无人车是个复杂的多系统融合，所以当前的标配 LiDAR、camera、IMU（六轴陀螺仪）都必须拥有。当你拥有了 camera 和 IMU 之后，做各种 VO（Visual Odometry）和最近的 VIO（Visual Inertial Odometry）SLAM 就是很自然的选择。前面提到的粒子滤波，还有各种版本的卡尔曼滤波，加上 graph 和关键帧的概念，都属于 SLAM 的范畴。作者曾详细介绍了 SLAM 的各种应用探讨和相对应的工程细节（<http://geek.csdn.net/news/detail/86807>），在此不再一一讨论，但值得注意的是，多传感器多信息源的融合只要算法正确并工程实现扎实，效果一定会比单一传感器好。举两个具体的例子：ICP 虽然后来有了很多改进（比如 point-to-distance 的测量改进，比如用 kd-tree 加速查找的改进），但 ICP 需要很好的初始化 **pose**，ICP 很容易掉入局部最优而搞不定全局最优，这时非常需要图像视觉的补充。图像的特征点提取特征描述的计算可以让匹配更精准，速度也不慢，但如果距离太远精度会下降，而且有很多实际的可以让图像视觉搞不定的情况（比如无人车或者别的物体的阴影，比如光照的变化，illumination-invariant 的特征非常难），所以作者认为多传感器多信息源的融合不是“让它变好”，而是“没你不行”。更明显的就是 GPS 在户外已经很可靠了，没有理由不用 GPS 先告诉无人车的大概位置。

12.4.2 导航与控制

跟踪和定位技术还是被动的感知方案，而真正意义的无人车也就是全自动驾驶而不是辅助驾驶是需要无人车自己智能地做路径规划的，我们简要地从技术层面谈一谈无人车的路径规划。

路径规划其实是一个范畴很大的话题，这里需要先做几个限定：第一个就是地图已知，如果地图未知没有“规划”可言，机器人或无人车如果完全对世界未知，那么问题实际是“SLAM+探索”；第二个是对无人车的领域来说，一般还是 2D 或 2.5D 的地图，而不是在 3D 地图上六个自由度运动规划（那是室内全自主无人机飞行），第三个就是路径规划默认的是无人车按照规划的路径每一步执行后的 **pose** 我们是默认准确的，也就是说，我们这里刻意地把定位和路径规划分开，但实际工程中这两者是紧密联系的，因为如果定位不准那么路径规划一定会受影响。

即使有了这几个设定之后，路径规划本身是有很多教科书版本的，而且种类繁多。这里简单谈两类中的比较有代表性又被广泛应用的两种。第一种是明确地寻找最好路径的搜

索 A*算法, 这种算法的核心理念是如果有最好的路径我要找到一条最好的路径。如果单位路径的成本 (cost) 不一样的话, 最好的路径不一定是最短的路径。如果有最好的, 一定要找到最好的路径。A*是搜索了所有的可能路径后选择了最好的, 而且运用了启发式算法决定最佳路径。A*的数据结构实现是 priority queue, 不停地在选取“最小成本”的节点扩建路径。

另一类路径规划是基于抽样 (sampling based) 的路径规划。简单地说, 从起点开始, 不知道最优路径是什么, 所以从起点开始随机抽样 (怎么随机也有讲究) 来扩建可能路径集, 但一个很重要的因素可以加速抽样, 就是障碍物的检测, 这个理解起来很直观, 如果有障碍物, 那么在障碍物方向再扩建路径没有意义。比较典型的算法是 RRT (Rapidly-exploring random tree), 但需要注意的是这种算法侧重的是要有效率地让树往大面积没有搜索过的区域增长, 那么实际运用中尤其是在如果有了一些启发式算法的情况下, 特别是在无人车的应用中, 实时的路径规划是很注重效率的, 所以要根据实际情况做优化, 这在研究上也有 RRT 的变种或两类算法的结合, 比如 A*-RRT。路径规划在无人车的工程实现一定是根据传感器的情况和地图质量来做实际算法的选择和调整, 比如地图到底有多准, 实时的各个传感器的数据质量如何, 在第一位永远是安全性的前提下, 更注重效率还是更注重绝对的优化等。

12.5 高精度地图的现状与结论

传统电子地图主要服务人类驾驶员, 而高精度电子地图主要服务无人驾驶系统。高精度电子地图的信息量与质量直接决定了无人驾驶系统的安全性、可靠性, 以及效率。与传统电子地图不同, 高精地图更精准 (厘米级), GPS 的准确度只有 5~10 米 (约 16~32 英寸); 高精地图更新更快, 并且包含了更多的信息 (语义信息)。由于高精地图的这些特性, 制作高精地图并不容易, 需要使用多种传感器互相纠正。在初始地图制作好后, 还需要对地图进行过滤以降低数据量去达到更好的实时性。在拥有了这些高精度的地图信息后, 无人驾驶系统就可以通过比对车载的 GPS、IMU、LiDAR 或摄像头的的数据精确地确认自己当前的位置, 并且进行实时导航。因为建设和其他变动, 高速公路地图需要定期更新, 无人车的地图也需要不断更新, 以便掌握路况变化。

截止目前为止, 日本最大的地图测绘公司 Zenrin 已宣布与 NVIDIA 合作, 研究如何简化利用人工智能绘制地图的流程。同时, 在高精地图提供商 HERE 新发展的合作关系

中, NVIDIA 在云中使用 MapWorks, 在车内使用 DriveWorks, 这两套地图开发工具的云端与地面相配合, 再加上 NVIDIA 为整个架构提供的 AI 引擎, 这样能够实现绘制和更新高清实时地图的目的。当然, 除此之外, NVIDIA 已经和 TomTom、百度建立了合作伙伴关系。在百度世界大会和 GTC 开发者技术大会(中国)上, NVIDIA 表示今后会联合百度开发从云到车的端对端的无人驾驶平台架构, 其中就包括会和百度在高精度地图绘制上做进一步优化提升。此前, 百度使用约 250 辆测绘车收集道路信息用以制作准确度在 5~10 米的传统导航地图及高精度地图, 其在中国绘制的高精度地图已经达到了厘米级精度, 包含交通信号灯、车道标记(如白线、黄线、双车道或单车道、实线、虚线)、路缘石、障碍物、电线杆、立交桥、地下通道等详细信息, 导航系统可以实现 95% 以上的路标和车道标记准确度。

12.6 参考资料

- [1] Thrun, S., 2002. Robotic mapping: A survey. Exploring artificial intelligence in the new millennium, 1, pp.1-35.
- [2] Levinson, J., Montemerlo, M. and Thrun, S., 2007, June. Map-Based Precision Vehicle Localization in Urban Environments. In *Robotics: Science and Systems* (Vol. 4, p. 1).
- [3] Konolige, K., 2004, July. Large-scale map-making. In *AAAI* (pp. 457-463).
- [4] Dellaert, F., Fox, D., Burgard, W. and Thrun, S., 1999. Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on* (Vol. 2, pp. 1322-1328). IEEE.

无人驾驶的未来

我们可以预测一个不远的未来，届时所有行驶的汽车都是无人车，我们将迎来一个更加安全、更加清洁环保的世界。得益于无人驾驶技术，未来我们的交通工具、行驶的道路，甚至是未来的世界都将变得更安全、更高效，极大地降低对石油燃料的消耗，减轻对环境的污染。本章，我们先从无人驾驶的商业前景、无人驾驶面临的发展障碍、无人车行业发展、全球化下的无人驾驶四个方面出发，分析未来无人驾驶的发展和即将面临的问题。最后，将给出无人驾驶发展的时间线，揭示在即将到来的未来二十年内无人驾驶的走势。

13.1 无人驾驶的商业前景

无人驾驶带来的商业潜力有多大？从本质上讲，无人驾驶和互联网的共同之处在于：它们都通过去人力化，降低了传输成本。互联网降低的是信息的传输成本，而无人驾驶则降低有形的物和人的运输成本。对比互联网已经产生的商业影响力，就可以想象无人驾驶的商业潜力。随着谷歌、Uber 和特斯拉这样的公司用事实不断展示技术上的进步，传统车厂已经越来越清晰地意识到，无人驾驶技术即将为汽车商业模式带来颠覆式的改变，这可能是自内燃机发明以来，汽车行业最重大的变化。

13.1.1 新的运输模式：TaaS 2.0

基于无人驾驶的发展，TaaS 2.0（运输即服务，Transportation as a Service）正在成为

业界探讨的热点,这里将 TaaS1.0 定义为有人驾驶,而无人驾驶则是 TaaS 2.0 时代。^[1]摩根士丹利公司在最近的一份报告中表示,实现汽车完全无人驾驶将极大地降低拼车成本,每辆车从目前的 1.50 美元一英里降至 20 美分一英里。借助无人驾驶,一箱德国啤酒从工厂出厂到中国的超市,在运输过程中可以不需要任何人工干预,全程自动化运输,中间会经过轮船运输、海关通关、高速公路运输和抵达城市等多个环节,其中的任务调度都可以在云端完成,这提供了端到端的运输解决方案。同样借助无人驾驶,从首都机场到上海的淮海路可以提前预约服务,中间也许会经过几次拼车,以实现运算效率最大化,就像手机基站切换那样,全程依然是无缝连接的。这一运输模式的变化对于传统车厂的影响是巨大的,一旦汽车从私人拥有变为共享运输工具,传统车厂的目标客户就将由个人消费者转变为 TaaS 运营商,正如华为和中国移动的关系一样。随之而来的是汽车的设计、销售都将发生根本性的改变是汽车厂商很难维持原来的强势地位。

13.1.2 无人驾驶的商业发展方向

因为安装了摄像头、雷达、激光雷达和人工智能系统,无人车的最初成本会很高,普通消费者难以接受。无人车可能会先进入特殊群体,比如企业和某些特殊行业将是早期消费群体。最有可能采纳无人车的行业包括约车、巴士、出租车、快递车辆、工业应用,以及为出行的老年人和残疾人士提供服务的行业。

1. 公共交通

无人车将成为公共交通系统的重要选择。百度计划几年后商业推广无人驾驶汽车,将首先在中国城市试运行。^[2]百度目前已经获得几个地方监管部门的批准,在事先确定的路线进行试验,希望在不远的将来推出这类车辆。一些城市还在考虑将某些街区划定为无人驾驶专区。在 30 或 40 个街区将不再出现人驾汽车和无人车同时存在的现象,无人驾驶出租车和共享出行车辆将提供全部交通服务。城市规划部门将进行区域优化,为无人车服务。

一项来自德克萨斯大学奥斯汀分校的关于分享无人车(SAV)的研究表明,“每辆 SAV 可以取代约 11 辆常规汽车,运营里程可以增加 10%以上”。^[3]这意味着,基于车辆分享的约车或出租车将缓解拥堵,大幅减少交通拥堵和环境恶化,因其方便性将广受消费者欢迎。

2. 快递用车和工业应用

快递用车和“列队”卡车将是另一个可能较快采用无人车的领域。^[4]在线购物和电子

商务网站快速兴起，给快递公司带来利好。人们喜欢在网上订购物品（如食品、货物和服务），几小时就能送货上门。中国电商 2015 年销售总额达到 5900 亿美元，很多产品承诺同日送达。这促进了电动车和卡车快递。2015 年，中国电商规模比 2014 年增长 33%。^[5] 卡车占美国机动车行驶里程的 5.6%，但是却占交通死亡事故的 9.5%。^[6] 因此，在经济效益和避免人员伤亡方面，无人车可以创造不少增加值。大型卡车成本通常超过 150000 美元，安装摄像头和感应器成本效益比较高，因为相比之下，小轿车的自身成本原本就很低，在无人驾驶初期因为成本的限制难以实现大规模推广^[7]。

3. 老年人和残疾人

在老年人和残疾人这两个消费群体中，无人车已经开始大规模应用。由于身体条件的限制和视力原因，这两类人都面临出行困难，因此智能车辆能给他们带来不少好处。美国老龄人口到 2050 年预计超过 8000 万，占总人口的 20%。届时，老龄人口将是目前的两倍，他们中有三分之一将面临出行困难。^[8] 中国也面临同样的情况。到 2050 年，中国老龄人口预计将占总人口的 33%。^[9] 而在日本，到 2060 年，65 岁及以上人群将占总人口约 40%。^[10] 残疾人的市场也很庞大。例如，在美国，约 5300 万成年人有残疾，占成年人人口的 22% 左右。约 13% 的美国成年人有出行障碍，约 4.6% 的成年人有视力障碍。^[11] 这些有关老年人和残疾人士的庞大数据为无人车提供了现成的市场。这两类群体都重视独立，无人车可以让他们自由出行，无须依靠朋友、家人。因此，老年人和残疾人对无人车的问世持积极态度。

13.2 无人车面临的障碍

无人车面临的障碍在技术层面包括恶劣天气、行车安全、隐私保护、基础设施不完善、频谱分配不足等；在社会层面包括事故追责、行车立法等。其中有些挑战是需要通过制度和社会行动才能逾越的障碍，这些领域都给无人车及其市场成功从技术层面带来了非常大的挑战。

13.2.1 恶劣天气

在恶劣天气里，无人车无法良好运行。大雨、大雪或大气雾霾遮挡道路标示和车道标记，因此增加事故风险。在这种情况下，无人车无法做出准确判断。Lyft 公司认为，无人车“在某些天气条件下或路况不佳时表现得不好。^[12]”另外，杜克大学的 Mary Cummings

教授特别强调恶劣天气对无人车的影响。“降水、雾和沙尘影响光雷达感应器、分散或阻挡激光束，干扰摄像头捕捉图像的能力。因此，车辆无法感知与其他汽车的距离，或者分辨停车标示、交通灯和行人。^[13]”

13.2.2 行车安全

安全是无人车行业的重要考虑因素。目前已经有车辆被黑客攻击、系统被破坏的报道。无人车依靠 V2V 的交流，以及 V2I 的连接。维护这些通道及电子邮件、电话、短信、上网和定位数据等乘客个人电子通信的安全至关重要。联网车辆面临的威胁包括黑客攻击、人为干扰、幽灵车或者其他恶意行为，如使用亮灯导致摄像头无法捕捉图像、雷达干扰或操控感应器等。上述每种行为都能扰乱通信，造成人工智能运算出错。他们的研究将“GNSS（全球卫星导航系统）欺骗和虚假信息列为最危险（即最可能或最严重）的攻击。”^[14] 操控这类信息给乘客带来风险，而且可能引发严重事故。网络安全专家已经证实能够远程攻击吉普大切诺基。《连线》杂志上刊登的一篇报道称，他们能够破坏车辆的转向、刹车、无线电、雨刷和车内环境控制系统，并且证实通过 Uconnect 软件能够很容易破坏车辆。这一例子说明，设计者需要高度重视车辆安全，避免不必要的风险。

13.2.3 隐私保护

无人驾驶隐私保护与安全不同，安全关注的是企图伤害司机和乘客的恶意行为，隐私则更关注数据的保护。无人驾驶过程中将收集大量关于人们位置、GPS 定位数据、网上购物（包括信用卡号码）和其他在联网汽车上进行的活动的信息，这些数据将创造新的商业模式。通过联网车辆，保险公司可以看到司机实际驾车的情况，实时风险和对情况的实时反应，并以此计算保费。汽车制造商和无人车服务企业的隐私政策一方面允许披露行车信息，用以“解决问题、评估使用和研究”。另一方面，一旦这些信息被非法转卖，匿名第三方极有可能将其用于营销甚至违法诈骗，损害消费者的利益。

因此，为了保护无人驾驶的隐私，缓解对无人车数据收集的担忧，首先应该提高网络安全标准，保证所有制造商能采取有效保护措施，尤其要提高无线网络下的数据加密保护。另一方面，需要提高设计安全性以减少攻击点、增加第三方测试、加强内部监督系统、设计分离架构以限制任何成功入侵造成的损害，以及不断更新升级安全软件以加强隐私保护的实时性。

13.2.4 基础设施不完善

基础设施不完善的问题在限制现有交通发展的同时,也扼制了无人驾驶的萌芽和起步。例如,在印度,高速公路和普通公路是主要挑战。印度的道路有 36% 是土路,而在中国,这一数字为 16% 左右。因此,在世界经济论坛的基础设施排行榜中,印度排第 87 位,远低于日本的第 6 位,德国的第 7 位,中国的等 46 位,泰国的等 48 位和巴西的等 76 位。^[15] 高速公路状况不佳给自动驾驶汽车带来挑战。车辆行驶需要可预测的路面和标示清晰的车道。在一次跨越国内多地的试驾过程中,Delphi 公司的工程师发现各地车道标示有很大差别。Glen De Vos 称,“自动驾驶车遇到有些路面标有宽的白条,有的则是窄的黄线。有些标示是新的,而有的模糊不清,有些道路还凹凸不平”。^[16] 由于这些道路标示不到位或工程质量不佳,半自动驾驶汽车或全自动驾驶汽车都无法顺利行驶。事故风险随之上升,电脑运算也很有可能做出错误判断。如果不加以解决,这将限制自动驾驶汽车的发展。^[17] 桥梁也是自动驾驶汽车面临的特殊问题。桥梁“提供的环境信息很少——如上面没有建筑物——因此车辆很难分辨确切位置”。^[18]

13.2.5 频谱不足

频谱不足也是无人驾驶推广的主要障碍。确定专用的频率范围是支持无人车发展的关键。有了具体的频段,无人车才能在任何天气或交通状况下行使。由于需要平衡联网速度和无线连接的可靠性,无人车和工业应用需要 6GHz 以下中等范围频谱,而这种频率的需求量大,很难确保无人车所需的可靠服务。电话断线让人厌烦,而如果无人车掉线则可能出人命。美国的车企大都支持专用短程通信(DSRC)系统。DSRC 是双向、中短距离无线通信协议,车辆之间可以相互沟通,发现和避开危险。装有 DSRC 的车辆在几百米的距离内每秒钟会多次播送准确信息,如位置、速度和加速情况。其他装有 DSRC 的车辆收到信息后会根据信息计算相邻车辆的运行轨迹,与自己的预计路径进行比较,再判断邻车是否有碰撞的威胁”。^[19] 联邦通信委员会于 1999 年规定 5.9GHz 波段的 75MHz 频谱为 DSRC 专有,为防碰撞技术的测试和应用打下了基础。但是,业界领袖希望给这一技术和联网汽车的其他功能留有足够的频谱。因为预计无人车将大量上市,应该为行业发展保留可用的频谱。

13.2.6 5G 通信尚未成熟

要实现无人驾驶的终极目标,不可避免地要解决网络延迟问题。当前的 4G 技术存在一定的信息延迟问题,信息延迟对无人车而言十分危险,也许一个刹车信号晚发出半秒就

可能造成一次严重的事故。随着 5G 技术的发展,信号延迟问题有望得到解决。这是因为,一方面 5G 技术能根据数据的优先级分配网络,从而保证无人车的控制信号传输保持较快的响应速度;另一方面,两车在行驶过程中,近距离直接数据连接的效率远高于绕道基站进行通信的效率。5G 技术将允许近距离设备直接通信,这样,可大大降低网络整体压力并降低平均延迟。只有解决了网络延迟问题,无人车技术难题才能得到进一步解决,未来的智能网联汽车才有机会完美实现车与人、车与车等范畴的智能信息交流共享。

因此,5G 网络被视为未来物联网、车联网等万物互联的基础。LTE-V 与 DSRC 是当前车联网的两大技术阵营,前者主要由中国企业推动,后者的发展比 LTE-V 成熟,是欧美等国车联网的主流技术。DSRC 即专用短程通信技术,是基于 IEEE 802.11p 标准开发的一种高效的无线通信技术,可提供高速的数据传输,并保证通信链路的低延时和低干扰,可实现小范围内图像、语音和数据的实时、准确和可靠的双向传输。较之 DSRC, LTE-V 的传输距离更远,信道更宽,同步性也更好。LTE-V 包括集中式 LTE-V-Cell 和分布式 LTE-V-Direct 两个工作模式。其中, LTE-V-Direct 模式可将车辆感知范围扩展到数百米的探测距离,与当前的车辆感知系统如雷达、光学摄像头的探测范围相比具有很大优势。由于通信技术的限制,当前的 LTE-V 版本属于 4.5G 技术,随着 5G 技术进一步发展,未来 LTE-V 将平稳演进到 5G。

13.2.7 事故追责

要实现全面发展,自动驾驶汽车行业必须解决法律责任的问题。目前,保险公司根据司机年龄、性别、经验等进行详细的风险评估。由于多数事故都是人为造成的,保险公司根据超速、酒驾、忽视道路标示或撞车等因素,确认事故的责任方。一旦自动或半自动驾驶发生行车事故,事故是谁的错?司机、控制自动装置的软件程序员还是制造硬件的汽车制造商?兰德公司的研究建议为无人车设置无过错保险,原因是无人车不容易受到人为失误影响。^[20]因此,无人车将更多责任从司机转移到制造商和软件设计者身上,彻底改变了行车事故的法律责任,完全颠覆建立在此基础上的法律体系和保险规则。

并且,无人车真正投向市场需要一个过程,人为驾驶汽车停止使用还需要很多年。以美国为例,美国有 2.62 亿辆汽车,平均寿命是 11.5 年。^[21]因此会出现 10 年甚至更长时间的新老汽车混合的复杂局面,造成事故的追责更困难且复杂。在中国,道路上有行人、自行车、低速车和高速车,全都混在一起,环境十分复杂。很多人都不按照一个标准骑车或开车。^[22]因此,在中国完成无人驾驶的事故鉴定要面临的情况更艰难。由此可见要实

现全面发展，自动驾驶汽车行业必须在清除技术障碍的同时，开始着手解决法律责任的问题。

13.2.8 行车立法

目前，公众对无人车的接受程度还处于中间状态。归根到底，公众必须要能接受（半）自动驾驶汽车，市场才能发展。和任何新兴技术一样，让消费者接受新模式和不同的行车方式，还需要时间，在此期间，对待向无人车的过渡也会有不同的看法。密歇根大学的美国公众调查显示，很多人仍然选择传统开车方式，46% 的美国人称不喜欢自动驾驶车，39% 的受众喜欢部分自动驾驶（半自动驾驶），16% 的人支持驾驶全自动无人车^[23]。当被问到具体装置时，95% 的人希望即使是全自动驾驶车也要有方向盘、油门和刹车板。37% 的人对乘坐全无人车表示十分担心，29% 的人称中度担忧，24% 的人轻度忧虑，10% 的受众毫无担心。

新技术出现的一个问题是通过测试其极限和努力，破坏其运行。如果高速行驶的大型设备遇到这种情况，后果会非常严重。无人车面临的危险多种多样，其中包括将激光照在汽车摄像头上破坏导航系统，攻击电脑代码，控制刹车和转向，将物体置于车前改变其运动，或发射电子信号改变其路线等。政策制定者应考虑制定法律，将针对无人车的恶意为定罪，通过对恶意行动立法，惩罚破坏无人车的行为。

13.3 无人驾驶产业

13.3.1 产业现状

自从无人驾驶进入公众视野，各大传统汽车厂商和高科技企业相继发布其在无人驾驶领域的研究成果，也相继推出了各自的（半）自动驾驶原型车，如图 13-1 所示。某些尚未动作的厂商，也展开了针对无人驾驶领域的大幅投入和行业合作，不得不说，现在正是无人驾驶产业蓬勃发展的时期。



图 13-1 已有（半）自动驾驶车原型（第一排从左至右：奥迪 A7、宝马 328Li、长安汽车、谷歌无人车；第二排从左至右：百度无人车、Uber、福特 Fusion 无人车）

1. Uber

Uber 的无人驾驶原型车采用了毫米波雷达、摄像头、激光雷达等传感器，并辅助以高精度地图。作为 2016 年 9 月公布的匹兹堡测试计划的一部分，Uber 推出了利用其无人驾驶技术改装的福特 Fusion 汽车，为了保证安全，每辆 Uber 无人驾驶专车上配备两名 Uber 工程师，一人坐在驾驶座上，随时准备在紧急情况下控制车辆，另外一人监控汽车的动态。Uber 收购了自动驾驶卡车创业公司 Otto。Otto 的自动驾驶卡车 10 月份行驶 120 英里（约合 193 公里），运送了 5 万罐啤酒。Uber 和沃尔沃还联合投资 3 亿美元开发自动驾驶汽车。

2. 谷歌

谷歌采取了不同于其他厂商的策略，将直接推出全自动无人车，而非逐步过渡的半自动驾驶功能。谷歌无人车所使用的传感器包括了毫米波雷达、激光雷达、摄像头，并且大量使用高精度地图。谷歌表示将努力在 2020 年完成无人驾驶的技术开发。截至 2016 年 11 月，谷歌的 60 辆无人车已经行驶超过 200 万英里。^[24] 该公司每月事故报告透明数据显示，7 年间只发生过 17 次小事故，没出现一例重伤。多数事故的原因是其他车辆的行驶难以预测，或者后方车辆追尾^[25]。为了保证行车安全，谷歌开发了备用的制动、转向和计算系统，以防备主系统失灵。另外还设计了软件，确保谷歌汽车“在其他司机的盲区之外”，远离车道上的摩托车，并且在绿灯亮起后停顿 1.5 秒，以避开闯红灯的车辆和行人。

现在谷歌无人驾驶项目已经拆分为单独的子公司 Waymo，Waymo 将和本田合作，双方计划在 2020 年前后实现高速公路上无人驾驶的实际运用。

3. 宝马

宝马以高精度地图结合激光雷达作为无人驾驶的研究方案,以“激光探测器+双目摄像头”作为量产方案。在上海车展上,宝马自动驾驶汽车配置了4个激光雷达,4个短距离77G毫米波雷达位于四个角保险杠;还有4个长距离77G,前面2个后后面2个;1个双目CAM在前,单目后4后1,4个Surrounding View CAM,都挨着短距77G毫米波雷达。

宝马在2015年7月上海创新日上曾以0~210km/h的车速实现了车速车道跟踪驾驶系统、全自动遥控泊车、防碰撞系统的演示。宝马与百度合作,以宝马3系作为平台,采用百度自动驾驶技术于2015年底实现了北京混合路测,从百度大厦—G7—五环路—奥林匹克森林公园往返,完成了途中的自动并线、超车、掉头、左转等操作。2016年,宝马、Intel和MobileEye将合作开发高度自动无人驾驶和全自动驾驶的解决方案和创新系统,预计在2021年量产无人驾驶电动车iNext,并最终非独家协议的方式,将该无人驾驶系统提供给业界,包括其他车厂。

4. 奥迪

奥迪的自动驾驶原型车传感器包括了激光雷达、ACC自适应雷达、超声雷达、Mobileye前置3D摄像头、前置红外摄像头,使用zFAS(TTTech+NVIDIA Tegra K1)作为中央驾驶辅助控制器。在2016年2月的柏林电影节上,奥迪已经实现了短距离的城市非封闭道路试验性的自动驾驶。奥迪计划2017年或者2018年量产可达到或接近达到L3的自动驾驶车辆。2021年推出L4无人驾驶车型。

5. 百度

百度无人车所使用的传感器包括了激光雷达、毫米波雷达、摄像头、红外、惯导,以及高精度地图。百度在2015年年底完成了城市环路及高速混合路况的全自动驾驶。在2016年7月,在乌镇景区推出了L4级别的无人驾驶服务。百度已经获批在加利福尼亚州测试其无人车,将于2021年开始大规模生产无人车。关于合作方面,百度与福特公司1.5亿美元共同投资激光雷达公司Velodyne LiDAR;百度与英伟达联合开发无人车平台。

6. 长安汽车

长安汽车现在有高速环境版和城区低速版两类无人驾驶原型车,使用的是“传感融合+高精地图”的技术路线。原型车上采用了激光雷达、毫米波雷达、单目摄像头等传感器。

长安汽车已实现了从重庆到北京的 2000 公里实际道路无人驾驶。长安计划 2020 年量产 L3 车型。

7. 特斯拉

特斯拉首席执行官埃隆·马斯克 (Elon Musk) 于 2016 年 10 月末宣布, 目前生产的车型包括 Model S、Model X 和未来的 Model 3, 在生产时采用了新硬件, 支持全自动无人驾驶技术, 一旦美国联邦政府开绿灯后, 这些汽车就可以升级无人驾驶功能了。此前, 特斯拉一直承诺到 2018 年推出具有全自动无人驾驶功能的汽车。值得注意的是, 2016 年 5 月特斯拉 S 型轿车在佛罗里达州发生致命车祸, 这是首起自动驾驶汽车致命车祸。事发时, 自动制动系统软件错把货车的白色车身当成晴朗的天空, 在卡车左转弯时未能识别。司机没有控制半自动装置, 结果轿车高速撞上了卡车, 冲向路灯杆, 司机当场死亡。^[26]

8. 沃尔沃

除与 Uber 在匹兹堡合作外, 沃尔沃还计划在中国提供“先进的自动驾驶体验”, 100 名志愿者将在公路上测试沃尔沃 XC90, 这是其 DriveMe 项目的一部分, 该项目计划于 2017 年在瑞典哥德堡和英国伦敦完成对 100 辆自动驾驶汽车进行的测试。

9. 尼桑

2016 年 7 月, 尼桑在日本发布了 ProPILOT——使汽车能在公路上自动行驶的自动驾驶功能。这一系统最终将登陆欧洲、中国和美国。尼桑将继续为 ProPILOT 增添自动驾驶功能, 直至 2020 年推出一款全自动无人车。

10. 福特

福特计划 2021 年将无人车用于汽车共享或打车服务。与谷歌的无人车相似, 福特无人车也没有方向盘、刹车或油门。福特计划 2017 年在欧洲测试其无人车, 把测试汽车数量由目前的 30 辆增加到 100 辆。福特 2016 年联合百度对激光雷达厂商 Velodyne 投资了 1.5 亿美元。

11. 通用汽车

通用汽车 2016 年宣布对 Lyft 投资 5 亿美元, 联合开发采用无人车的打车服务。通用汽车 2016 年 3 月还以 10 亿美元收购了自动驾驶汽车创业公司 Cruise Automation, 增强在自动驾驶汽车开发方面的实力。

12. 丰田汽车

丰田宣布正在密歇根大学建立“自动驾驶汽车研究基地”，帮助丰田建设原始模型实验室，在模拟路况中进行低速车辆测试。

13.3.2 产业发展

目前，无人驾驶发展的格局呈现两种趋势，如图 13-2 所示，传统汽车公司更倾向于渐进式自动化，而以特斯拉、Google、Baidu 等为代表的科技公司是以人工智能进入完全自动无人驾驶，具有颠覆式的色彩。他们短期内发展的目标也各有侧重，传统车企以渐进自动驾驶的方式逐步提升驾驶体验，而科技类公司直接跨越到以无人驾驶完全替代传统驾驶。前者依靠的是长期积累的整车经验和在自动控制领域的核心优势，后者则是借由人工智能的大力发展整合传感器、感知算法、计算平台等技术以实现跨越式的发展。他们在营利模式上也有所区别，传统汽车公司依然着重整车销售和售后市场的服务，而科技公司更注重内容资源、平台服务和共享经济带来的新商业模式。

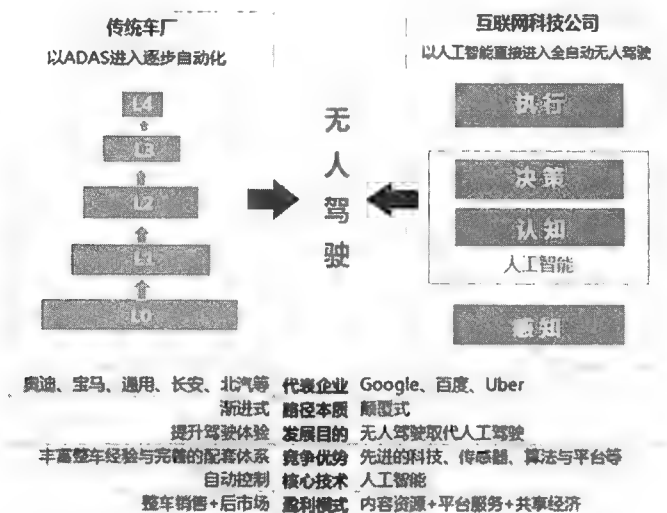


图 13-2 各类企业在无人驾驶行业的布局

来自中信证券的报告显示，预计至 2025 年，科技型企业将在智能汽车领域分得 40% 的市场，而传统车企将坚持 60% 的市场。科技型企业直接跨越到高自动化无人驾驶，作为行业新进入者并无历史“包袱”，可以实现跨越式发展。此外，科技型公司在数据融合、高精地图等方面具有技术优势，这些既有的优势将助力高速发展。通过实现布局无人驾驶，

科技型公司可以将汽车变成下一个“互联网入口”。也有预测认为，中短期科技型企业依然无法撼动传统车企占据的过半市场份额。传统车企从辅助驾驶开始逐步提升，既有品牌与用户优势，且造车门槛较高，因此传统整车厂仍保有全产业链优势，产品安全可靠更高，且消费者对传统品牌认可度较高，汽车产业不会被科技型公司完全颠覆。当然，科技企业与传统车企的合作越来越多，他们之间的联姻，包括合作与投资，也许会是激烈竞争的另外一个结果。

13.4 全球化下的无人驾驶

本节将探讨在全球化的背景下无人驾驶在各个国家和地区带来的预算、政策和监管等具体问题和相应对策。

13.4.1 无人驾驶在中国

世界经济论坛的一项调查发现，“75%的中国人愿意乘坐无人车。”^[27]这种观点也得到了罗兰贝格公司单独调查的印证。后一项调查显示，“96%的中国人愿意考虑每天都乘坐无人车，远高于美国和德国的58%。”^[28]相比其他国家和地区，中国对无人驾驶的接受程度非常高，因此大规模商业化之后，无人驾驶的推广也将更加迅速。另一方面，在中国大部分无人车的监管规范的制定都在国家一级，其自上而下的做法也能简化监管规则和程序。这也是无人驾驶将在中国得到领先发展的一大优势。

因此，无人驾驶在中国的关键是要制定国家层面的无人车政策框架。中国有多个部委负责无人驾驶技术的监管（其中存在责任交叉和重复，以及碎片化的问题）。国家质量监督检验检疫总局（负责产品召回）、工业和信息化部（负责制定产业政策）、交通部（制定交通行业发展规划）、公安部（负责车辆登记、牌照管理和交通安全监督）、国家测绘地理信息局（负责执行地图信息采集规定）都有管辖权。还有其他部门负责环境保护、回收利用、商务和金融。因此需要进一步明确监管部门和监管方式。

目前，中国已经启动了LTE-V无线技术和频率划分研究试验工作，华为主推的Polar码也被选为5G eMBB（增强移动宽带）场景的控制信道编码方案，这表明，中国已经开始全面布局基于5G的自主车车通信技术，以此推动无人驾驶的发展。

13.4.2 无人驾驶在欧洲

欧洲国家发展无人驾驶乏力的原因之一是没有很多实力雄厚的本土技术公司。欧洲在无人驾驶相关的人工智能、网络、绘图、芯片、感应器、设备和服务方面需要依靠外部企业。美国的谷歌和中国的百度等大型技术企业之所以能快速进军交通领域，其中一个原因是，它们有机会将搜索引擎技术开发获取的信息处理专长和快速学习能力应用到无人驾驶领域。若想获得无人车领域的竞争能力，奥迪、宝马（与英特尔合作）、大众、戴姆勒、奔驰和沃尔沃等欧洲车企应该培养人工智能、深度学习、大数据分析、高精度测绘等领域的人才，这些技术对未来汽车发展至关重要。

另一反面，欧盟还应确保其数据保护规定不会对车辆和行人的移动和位置信息带来过分限制，从而抑制无人驾驶所需的高精度地图的绘制。欧盟一贯在数据收集和分析领域采取限制性立场。例如，欧盟法规限制谷歌等企业收集路况数据和绘制街景地图。就此，欧盟已经多次因此处罚谷歌，索要谷歌的数据，并对收集的资料进行限制。^[29] 最近通过的《欧盟数据保护总则》严格限制人工智能和机器学习的使用。这些规定导致很难将人工智能和高精度测绘纳入无人驾驶汽车之中，没有包含地理编码数据和利用此类信息的深度学习，无人驾驶技术很有可能将在欧洲停滞。

13.4.3 无人车在日韩

相比中美，日韩政府和车企一直对无人车持谨慎态度。日韩车企投入了大量资源，它们关注其他国家的情况，也在实施试点项目。但是它们对是否将无人车列为发展重点一直持谨慎态度。2016年4月，丰田公司宣布正在密歇根大学建立“自动驾驶汽车研究基地”，其目标是建设原始模型实验室，在模拟路况中进行低速车辆测试，帮助公司在人工智能和高清数据绘图领域迎头赶上。^[30] 虽然做了这笔投资，公司领导却宣布“丰田相信丰田汽车绝不会发展到司机无须手握方向盘的无人驾驶状态”。^[31]

日本政府2015年发布指南，确定2020年在车内安装半自动装置，2025年启动全自动无人车。100韩国车企起亚和现代已确定于2030年完成全自动无人车运行的目标。但是它们的计划远远慢于美国和中国，在这两国，半自动驾驶汽车已经上路运行，全自动无人车也将于2020年投入使用。

13.4.4 无人驾驶在美国

无人驾驶在美国的主要挑战在于解决监管碎片化的问题，因为五十个州在许可、车辆

标准、监管和隐私保护等领域的政策不同。目前,车企(如福特和通用)和软件开发商在多个州都面临规则和监管相冲突的问题。这会影响创新,因为企业在制造无人车时,要满足全国或国际市场的需要。

13.5 无人驾驶发展对策

13.5.1 制定国家政策

政府可以通过监管的方式加速或放缓向无人驾驶的过渡。最重要的监管任务应该是“支持道路测试和澄清控制和适当控制的含义。”目前,关于“谁或什么在控制”及什么是适当控制还存在不确定性。^[32] 报告的作者呼吁制定道路试验、司机规则、控制期望和法律责任的^[32] 国家指南。

以 ADAS 技术为例,如表 13-1 所示,目前世界各国政府已经逐渐加强法规,都将 ADAS 主动安全技术纳入安全评分体系,要求在新车型中配备能够有效加强汽车安全性的 ADAS 系统,如 LDW 车道偏离警示、FCW 前碰预警系统、AEB 紧急制动系统等。中国国家汽车工业协会在 2016 年发布的十三五规划中,提出了具有辅助自动驾驶功能 50% 新车渗透率,具有部分自动驾驶功能达到 10% 新车渗透率的目标。2016 年 12 月 14 日,美国交通部发布了 V2V 的新法规,进入了 90 天公示期,法规强制要求新生产的轻型汽车安装 V2V 通信装置,这是一个里程碑式的进步。这无疑会推动自动驾驶技术在产业化方面的快速进展。

表 13-1 各国关于 ADAS 的立法

地区	机 构	时 间	主要内容
美国	NHTSA	2011 年	在车辆评分系统中加入 FCW(前方碰撞预警),要求车辆配套 LDW(车道偏离警示)和 FCW
		2015 年	2018 年起,五星安全必须配备 AEB(自动紧急刹车)
		2016 年	占美国汽车 99% 以上的汽车生产商同意自 2022 年起在轻型乘用车和卡车上配备 AEB
	IIHS	2015 年	对于安装 FCW 等碰撞规避系统的车辆提高其安全等级
欧洲	NCAP	2013 年	大型上用车必须配备 AEB
		2014 年	Vehicle-AEB(对车—自动紧急刹车)成为新车评估测试的必要条件
		2014 年	五星商用车至少配备一项主动安全技术

续表

地区	机构	时间	主要内容
日本		2016 年	Pedestrian AEB（对人—自动紧急刹车）成为新车评估测试的必要条件
		2017 年	4 分评级车辆必须配备主动安全
	MLTI	2014 年	将 AEB 纳入评分系统
		2016 年	强制车辆安装 AEB
澳大利亚	NCAP	2012 年	只有安装了 AEB 和 LDW 的车辆才能达到五分评级
中国	汽车工业协会	2016 年	具有辅助驾驶功能 50%新车渗透率，具有部分自动驾驶功能 10%新车渗透率
	NCAP	预计 2018 年	在安全评分系统中加入 AEB、LDW、FCW、PDS 等

13.5.2 改善高速公路基础设施

基础设施不完善是无人驾驶发展的重大技术障碍，因此改善高速公路应该是发展自动驾驶汽车的重点。无人车载摄像头如果无法识别车道标示，3D 高精度地图也无用武之地。另外，如果缺少安装智能路灯，在视线受到限制的情况之下，摄像头将很难工作。另一方面，现在的交通信号灯效率低，没有考虑到车流信息。如果交通灯是动态的，可根据车流量调整间隔，将可以缓解道路交叉口的拥堵，更好地与无人驾驶相配合。因此，政府需投资建设针对无人车的高速公路基础设施。

13.5.3 允许道路测试和精准道路绘图

现有各国的法规都要求车辆行驶时司机必须在车内，而且双手置于方向盘上，这明显给全自动无人车的推广带来了限制。以日本为例，现在要求国内所有汽车都由真人驾驶。车企需要特殊许可才能进行道路测试。国家警察厅的规定要求“所有公共道路上的测试都要有司机坐在方向盘后面，”这也就限制了全自动无人车的测试和发展^[33]。因此，政府应该取消全国范围内对于道路测试的禁止，开放道路进行车辆测试。道路测试是自动驾驶汽车自验证的最佳方式。从长期看，这是保证无人驾驶行车安全的最佳方式。成功的试验项目也让公众更相信自动驾驶汽车的安全性和可靠性。

同时，也需减少公路地图发展方面的限制。精确的地图关乎全自动无人车的未来发展。现有技术可以将线路图的误差降低到几厘米。出于安全考虑，中国政府规定公共地图的精确度不能超过 50 米（165 英尺）。因此，很难为无人自动驾驶汽车绘制精确的 3D 高精度

地图。如果达不到这一精确度,车辆无法安全行驶。企业需要获得国家测绘和地理信息局的特殊许可才能收集路况和桥梁高度、宽度等数据。企业还禁止收集军事管理区周围的任何道路信息。这给行业创新带来不必要的负担,也给行业收集所需信息带来难题。在欧洲,欧盟法规限制谷歌等企业收集路况数据和绘制街景地图。以德国为例,谷歌在采集公共道路街景数据之前,必须告知公众。除此之外,德国还限制公司保留私人住宅周围街道和高速公路图像数据的时间。人们也可以选择排除在数据采集之外,已经有 1/3 的德国人这样做了。^[34] 这些规定都限制了公路地图绘制的准确性和更新地图的能力。

13.5.4 制定技术标准

制定技术标准对自动驾驶汽车至关重要。技术标准现在更多地变成一个国家贸易保护的壁垒,成为产业竞争的制高点。就无人驾驶而言,技术标准可以帮助高精度地图的测绘,规定高精地图在测量、分析和精确度等方面的要求。这样可以在政府规定模糊的领域进行澄清。车辆无线技术也是一个需要确定技术标准的领域。例如,中国在 2016 年下半年发布了《推进“互联网+”便捷交通促进智能交通发展的实施方案》,明确提出“结合技术攻关和试验应用情况,推进制定人车路协同(V2X)国家通信标准和设施设备接口规范,并开展专用无线频段分配工作”的标准制定工作。从目前的情况来看,LTE-V 极有可能确定为中国在无人车通信方面的标准,而不会使用欧美的 DSRC 标准。

13.6 可预见的未来

13.6.1 无人驾驶的黎明:现在—2020 年

现今已有无数人投身于无人车领域,长期积累的科研成果及工程进步都致力于将无人驾驶在 2020 年成为现实。人工智能是无人驾驶的核心,但是无人车是一个相当复杂的工程系统,需要众多技术的融合与精确配合,其中包括以下几方面。

(1) 算法:算法部分包括传感,用来从采集到的传感器原始数据中提取有意义的信息;定位,用来精确地控制无人车的行驶方向;感知,用来理解车辆的周边环境,并为车辆的出行与到达提供安全可靠的规划。

(2) 客户端系统:这部分由操作系统和硬件系统组成,将配合算法部分以满足实时、可靠、安全、绿色能耗的要求。

(3) 云平台：这部分提供离线的计算和存储功能以支持测试不断更新的算法、产生高精度的地图及大规模的深度学习模型训练。

13.6.2 混合模式的时代：2020—2040 年

我们即将进入混合模式的时代：2020—2040 年。在这 20 年间，传统的人为操控汽车及无人车将共存。考虑到每一辆机动车的使用寿命是 10~15 年，我们可以预见这一人为驾驶/无人驾驶的情况将持续至少 20 年。早期的无人车被设计为能够理解并能处理传统的面向人为驾驶的交通系统。随着无人驾驶的普及，交通系统将逐渐演化为对无人车更友好的模式，交通灯、Lanes 及 Stop Sign 将进一步装备在路感应器，以更好地辅助无人驾驶。此外，无人车之间的通信将急剧增加，因此能够更好地完成行驶过程中车辆的动态协调。在这一背景下，持续产生的大量数据将推进 AI 算法的持续修正与进步。

13.6.3 无人驾驶时代：始于 2040 年

到 2040 年，预计所有的汽车将完全转变为无人驾驶模式，今后人为驾驶会成为一件罕有的事情，甚至于由于缺乏足够的安全性被判定为非法行为。届时，我们将迎来全新的交通生态系统，在这个生态下，所有的车辆都是集中控制模式。基于无人驾驶的自动交通运输将像供电、供水一样，成为日常生活中的基础设施。得益于改进的导航系统及传感器对路面和车辆老化状况的检测，传统汽车行驶中的交通事故由现在的每年超过百万起降低至几乎零事故。当然，正是因为无人驾驶驱动的公共交通对资源的有效共享与分配，整个城市的交通系统只需要较少量的汽车便可以正常运行。一方面能源的使用效率将被极大地提高，另一方面将大规模地替代使用新能源，因此，传统化石燃料造成的空气污染将被大规模地降低。我们迫切地期待着无人驾驶的到来。

13.7 参考资料

- [1] GOOGLE, UBER, AND THE EVOLUTION OF TRANSPORTATION-AS-A-SERVICE, www.stratechery.com.
- [2] 2016 年 4 月 23 日 phys.org 上的文章 “Chinese Firms Accelerate in Race Toward Driverless Future”, 作者 Julien Girault.
- [3] Daniel Fagnant 和 Kara Kockelman 在 2014 年 1 月交通研究理事会第 93 次年会

- 提交的文章 “The Travel and Environmental Implications of Shared Autonomous Vehicles Using Agent-Based Model Scenarios” , 第 1-13 页.
- [4] 2016 年 7 月 25 采访兰德公司 Karlyn Stanley.
 - [5] 2016 年 5 月 30 日《纽约时报》文章 “Beijing’ s Electric Bikes, the Wheels of E-Commerce, Face Traffic Backlash”, 作者 Chris Buckley.
 - [6] 2016 年 5 月 17 日《纽约时报》文章 “Want to Buy a Self-Driving Car? Trucks May Come First”, 作者 John Markoff.
 - [7] 2016 年 5 月 17 日《华尔街日报》文章 “Autonomous-Driving Venture Targets Heavy Trucks”, 作者 Mike Ramsey.
 - [8] 美国统计局 2014 年 5 月报告 An Aging Nation: The Older Population in the United States”, 作者 Jennifer Ortman, Victoria Velkoff 和 Howard Hogan。美国交通部报告 “Beyond Traffic, 2045: Trends and Choices”, 第 11 页.
 - [9] 布鲁金斯学会学会技术创新中心 2014 年报告 “mHealth in China and the United States”, 作者 Yu Xiaohui, Han Han, Du Jiadong, Wei Liurong, Li Cheng, Zhang Xueli, Li Haihua, Huang Ying, Sun Ke, Li Na, Darrell West 和 Joshua Bleiberg。第 9 页.
 - [10] 彭博技术 2015 年 10 月 28 日文章 “Japan’ s Carmakers Proceed With Caution on Self-Driving Cars”, 作者 Jie Ma.
 - [11] 2015 年疾控中心报告 “53 Million Adults in the US Live with a Disability” .
 - [12] 2016 年 7 月 5 日采访 Lyft 公司 Rob Grant.
 - [13] TR News 2014 年 5-6 月文章 “Who Is In Charge? The Promises and Pitfalls of Driverless Cars”, 第 28 页, 作者 Mary Cummings 和 Jason Ryan.
 - [14] 2015 年 4 月第 16 卷 IEEE Transactions on Intelligent Transportation Systems, 第 554 页, 作者 Jonathan Petit 和 Steven Shladover.
 - [15] 波士顿咨询公司 2014 年 11 月报告 “Make in India: Turning Vision Into Reality”, 第 20 页, 作者 Arindam Bhattacharya, Arun Bruce, Anirban Mukherjee.

- [16] Delphi 公司的 Glen De Vos 2016 年 3 月 15 在参议院商业、科学和技术委员会听证会上的证词，第 6 页。
- [17] 2016 年 1 月 24 日彭博新闻 “Baidu Enters the Global Race for Driverless Car Domination” .
- [18] 2016 年 8 月 18 日《彭博新闻周刊》文章 “Uber’s First Self-Driving Fleet Arrives in Pittsburgh This Month”，作者 Max Chafkin.
- [19] Sandy Lobenstein 2015 年 11 月 18 日在众议院监督和政府改革委员会的证词，第 5 页。
- [20] 兰德公司 2016 年报告 “Autonomous Vehicle Technology: A Guide for Policymakers”，作者 James Anderson, Nidhi Kalra, Karlyn Stanley, Paul Sorensen, Constantine Samaras 和 Oluwatobi Oluwatola, 第 116 页。
- [21] Delphi 公司的 Glen De Vos 2016 年 3 月 15 在参议院商业、科学和技术委员会听证会上的证词，第 6 页。
- [22] 2016 年 4 月 3 日《纽约时报》文章 “China’s Companies Poised to Take Leap in Developing a Driverless Car”，作者 John Markoff 和 Paul Mozur。
- [23] 密歇根大学世界可持续交通研究所报告 “Motorists’ Preferences for Different Levels of Vehicle Automaton: 2016”，作者 Brandon Schoettle 和 Michael Sivak.
- [24] 2016 年 6 月 21 日采访谷歌 George Ivanov.
- [25] 2015 年 7 月 17 日 CBS 新闻 “Google Driverless Car Involved in First Injury-Causing Accident” .
- [26] 《纽约时报》2016 年 7 月 12 日文章 “As U.S. Investigates Fatal Crash, Company Defends Autopilot System”，作者 Bill Vlasic and Neal Boudette。2016 年 7 月 29 日文章 “Tesla Faults Brakes, but Not Autopilot, in Fatal Crash”，作者 Neal Boudette.
- [27] 密歇根大学世界可持续交通研究所报告 “Motorists’ Preferences for Different Levels of Vehicle Automaton: 2016”，作者 Brandon Schoettle 和 Michael Sivak。
- [28] 路透社 2016 年 4 月 23 日文章 “China’s Roadmap to Self-Driving Cars”。《中国日报》2016 年 6 月 22 日文章 “Officials Want to Open Way for Autonomous

Driving” .

- [29] 《纽约时报》2013 年 4 月 23 日文章 “Germany’s Complicated Relationship with Google Street View”，作者 Claire Miller 和 Kevin O’Brien.
- [30] 《底特律新闻》2016 年 4 月 7 日文章 “Toyota to Launch Ann Arbor Autonomous Car Research Base”，作者 Michael Martinez.
- [31] 《底特律新闻》2016 年 4 月 7 日文章 “Toyota to Launch Ann Arbor Autonomous Car Research Base”，作者 Michael Martinez.
- [32] 澳大利亚国家交通委员会 2016 年 5 月报告 “Regulatory Options for Automated Vehicles”，第 8-9 页.
- [33] 《汽车与自行车》杂志 2016 年 5 月 28 日文章 “Japan Won’t Allow Testing of Fully Autonomous Cars On Public Roads”，作者 Sameer Contractor.
- [34] 《纽约时报》2013 年 4 月 23 日文章 “Germany’s Complicated Relationship with Google Street View”，作者 Claire Miller 和 Kevin O’Brien.

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），您即可享受以下服务：

- **提交勘误：**您对书中内容的修改意见可在【提交勘误】处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **与作者交流：**在页面下方【读者评论】处留下您的疑问或观点，与其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/31355>



作者介绍



刘少山, PerceptIn联合创始人。加州大学欧文分校计算机博士。现在PerceptIn主要专注于机器人的核心SLAM与深度学习技术, 以及其在智能硬件上的实现。在创立PerceptIn之前, 在百度美国研发中心主要专注于百度无人车系统架构与产品化、深度学习, 以及异构计算平台的架构与开发。



唐洁, 华南理工大学计算机科学与工程学院副教授。唐洁博士现主要从事面向无人驾驶和机器人的大数据计算与存储平台、面向人工智能的计算体系架构、面向机器视觉的嵌入式系统研究。



吴双, 依图科技研究科学家, 依图硅谷研发中心负责人。原百度研究院硅谷人工智能实验室资深研究科学家, 原百度美国研发中心高级架构师。美国南加州大学物理学博士, 加州大学洛杉矶分校博士后, 研究方向包括计算机和生物视觉, 互联网广告算法和语音识别, 曾在NIPS等国际会议中发表文章。



李力耘, 百度美国研发中心无人驾驶高级架构师。本科毕业于清华大学电子工程系, 后获得美国纽约大学计算机专业博士学位。加入百度后从事移动推荐、转换广告、图片变形、无人车决策规划等多个项目。目前在百度无人车部门负责无人车行为预测方向的系统架构及算法优化。拥有多项国际专利, 并已递交30余项无人车决策预测相关专利申请。



博文视点Broadview



新浪微博
weibo.com

@博文视点Broadview



责任编辑: 郑柳洁
封面设计: 侯士卿

欢迎投稿
邮箱: zhenglj@phei.com.cn
微信号: Alinamercy

000000

上架建议: 人工智能

ISBN 978-7-121-31355-4



定价: 59.00元